HOLOMAKERS PROJECT

Motivating secondary school students towards STEM careers through hologram making and innovative virtual image processing practices with direct links to current research and laboratory practices

Erasmus+ KA2 2017-1-PL01-KA201-038420

Output 1

HOLOMAKERS – Οδηγός αναφοράς για την ολογραφία για εκπαιδευτικούς

Lead Partner: WUT

Authors: Artur Sobczyk (WUT)

Circulation: *Restricted/Public*

Version: *01*

Stage: *Final*

Date: *2018*

Contributions

Karol Kakarenko, Warsaw University of Technology
Rene Alimisi, EDUMOTIVA
Chrysanthi Papasarantou, EDUMOTIVA

Declaration

This report has been prepared in the context of the HOLOMAKERS project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

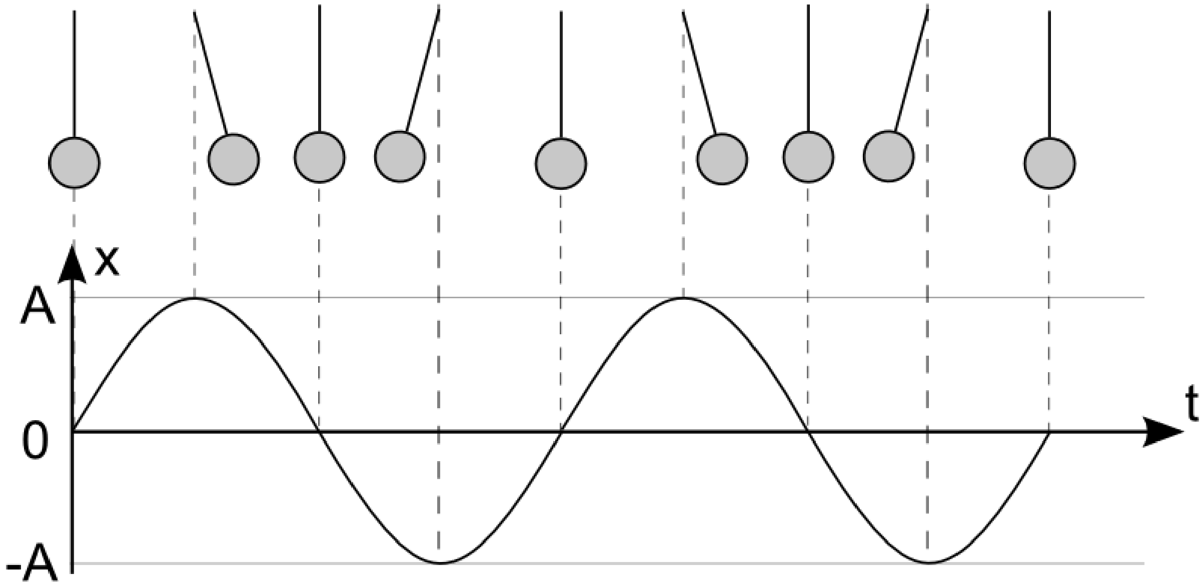
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Κεφάλαιο 1. Κατανοώντας τα κύματα

1.1. Τα κύματα στη φυσική

Στη φυσική, ως κύμα νοείται η διάδοση μιας διαταραχής/ταλάντωσης σε ένα μέσο ή στον χώρο. Αν, για παράδειγμα πετάξουμε μια πέτρα στο νερό, θα δημιουργηθεί ένα κύμα στην επιφάνεια του νερού, το οποίο θα διαδοθεί, σε κάποια απόσταση, από το σημείο όπου έπεσε η πέτρα. Σε αυτή την περίπτωση, η ενέργεια κρούσης/πρόσκρουσης θα μετατραπεί σε μια σειρά ταλαντώσεων του μέσου (του νερού, στο παράδειγμά μας). Ένα άλλο παράδειγμα αποτελεί το ηχητικό κύμα. Σε αυτή την περίπτωση το κύμα ταξιδεύει λόγω της ταλάντωσης του αέρα.

Επίσης, μπορούμε να ονομάσουμε κύμα μια ταλάντωση που περιορίζεται στον χώρο. Για παράδειγμα, ένα κινούμενο εκκρεμές προκαλεί την ταλαντευόμενη κίνηση να επιστρέφει, στο ίδιο σημείο, κάθε νέα χρονική στιγμή. Σε αυτή την περίπτωση, το κύμα είναι το πλάτος μιας αιώρησης του εκκρεμούς, η οποία μεταβάλλεται στον χρόνο. Στην εικόνα 1 απεικονίζεται η μεταβολή της ταλάντωσης τους εκκρεμούς, στον χρόνο. Αν [-A, A] είναι η περιοχή εκτροπής του εκκρεμούς, μπορούμε να ορίσουμε ως A το πλάτος (amplitude) της ταλάντευσης.



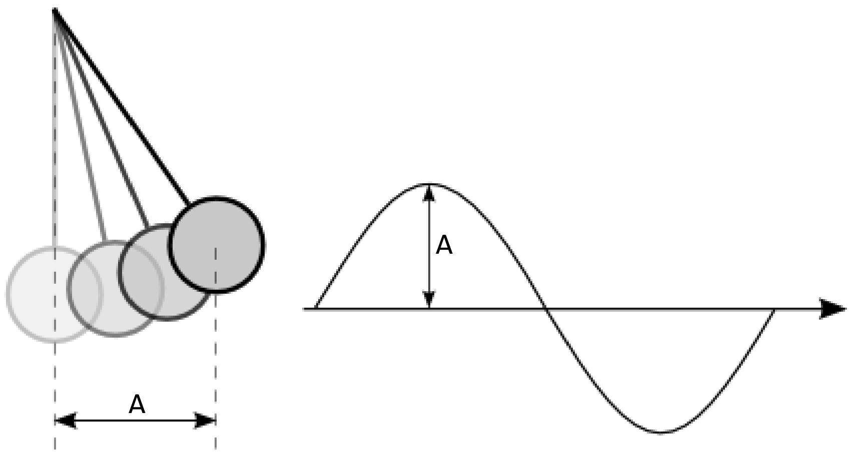
Εικ. 1 Η μετατόπιση του εκκρεμούς σε διαφορετικούς χρόνους

Πολύ συχνά, η περιγραφή των ταλαντώσεων μπορεί να γίνει μέσω της ημιτονοειδούς λογικής (sine function). Αυτό συμβαίνει τόσο στην περίπτωση του εκκρεμούς, όσο και στην περίπτωση ενός βάρους που κρέμεται σε ένα ελατήριο, ή ακόμα και στην περίπτωση μιας χορδής ενός μουσικού οργάνου. Αυτός είναι ο βασικός τύπος ταλάντωσης. Τέτοιου τύπου ταλαντώσεις ονομάζονται **αρμονικές**. Όπως θα μάθουμε αργότερα, οποιοσδήποτε άλλος, πιο πολύπλοκος τύπος, ταλάντωσης αποτελείται από αρμονικές ταλαντώσεις. Εν κατακλείδι, ως αρμονικό κύμα νοείται αυτό που μπορεί να περιγραφεί μέσω της χρήσης της ημιτονοειδούς λογικής. Στο επόμενο κεφάλαιο θα μάθουμε τις βασικές ιδιότητες που χρησιμοποιούνται στην περιγραφή των κυμάτων.

1.2. Τα χαρακτηριστικά των κυμάτων

1.2.1. Πλάτος

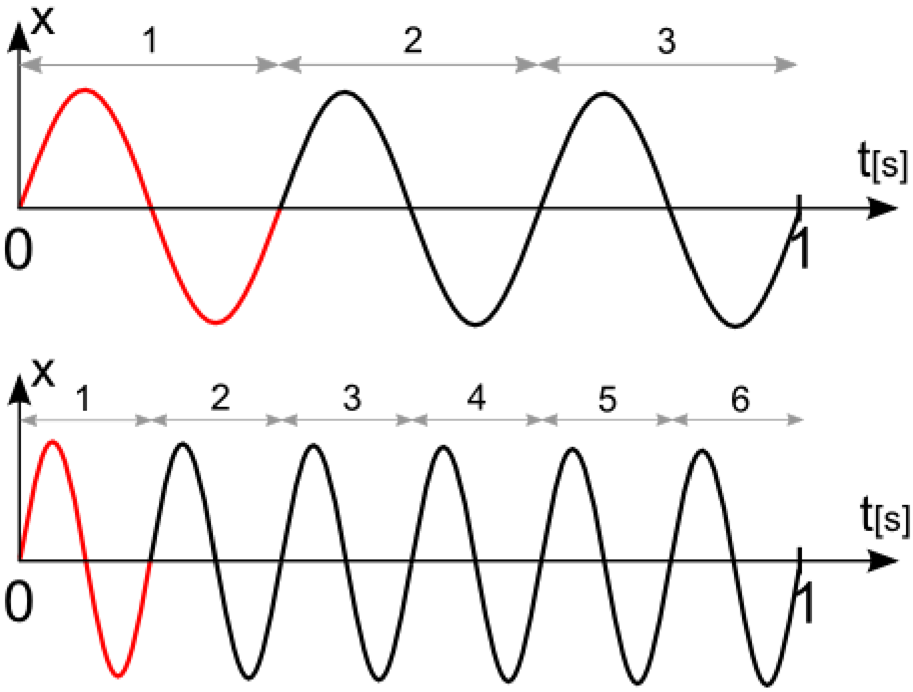
Ως πλάτος ενός κύματος ορίζεται η απόσταση της μεγαλύτερης εκτροπής από την θέση ισορροπίας. Στην περίπτωση του εκκρεμούς, το πλάτος της ταλάντωσης είναι η μεγαλύτερη εκτροπή του (βλέπε εικόνα 2). Η μέγιστη τιμή κύματος ονομάζεται κορυφή (crest) και η ελάχιστη κοιλάδα (trough).



Εικ. 2 Το πλάτος της ταλάντωσης του εκκρεμούς

1.2.2. Συχνότητα

Η συχνότητα μας ενημερώνει για το πόσες ταλαντώσεις έχουν γίνει σε μια μονάδα χρόνου (εντός ενός δευτερολέπτου). Η μονάδα της συχνότητας είναι το Hz (Hertz). Για παράδειγμα, 10Hz ισούνται με 10 ταλαντώσεις το δευτερόλεπτο. Η εικόνα 3 παρουσιάζει δύο κύματα με διαφορετικές συχνότητες. Η κόκκινη γραμμή περιέχει μια πλήρη ταλάντωση. Η συχνότητα του πρώτου κύματος είναι 3Hz (3 ταλαντώσεις το δευτερόλεπτο) και του δεύτερου 6 Hz (6 ταλαντώσεις το δευτερόλεπτο).



Εικ. 3 Συχνότητα των κυμάτων

1.2.3. Περίοδος

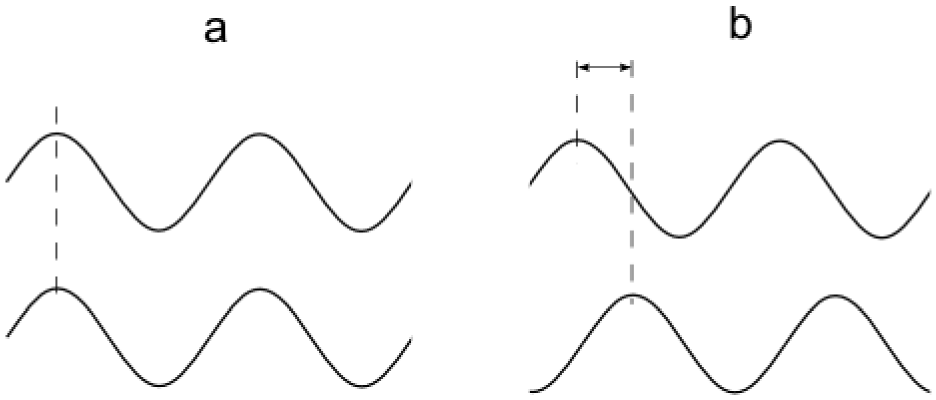
Η περίοδος ενός κύματος συνδέεται άμεσα με τη συχνότητά του. Είναι ο χρόνος (μετρημένος σε δευτερόλεπτα) στον οποίο μια πλήρης ταλάντωση συμβαίνει (και που σημειώνεται με κόκκινο χρώμα στην εικόνα 3). Η σχέση εξάρτησης περιόδου κύματος (T) και συχνότητας (*f*) είναι $T = \frac{1}{f}$. Έτσι, η περίοδος του πρώτου κύματος στην εικόνα 3 είναι $\frac{1}{3}s$, ενώ η περίοδος του δεύτερου ισούται με $\frac{1}{6}s$.

1.2.4. Μήκος κύματος

Το μήκος ενός κύματος μπορεί να καθοριστεί, εφόσον αυτό διαδίδεται στον χώρο. Σε αυτή την περίπτωση λοιπόν, το μήκος είναι η απόσταση στην οποία το κύμα θα διαδοθεί στη διάρκεια μιας περιόδου. Το μήκος κύματος (λ) δίνεται από την εξίσωση $\lambda = vT$, όπου v είναι η ταχύτητα διάδοσης του κύματος και T η περίοδός του.

1.2.5. Φάση

Η φάση καθορίζει σε ποιο τμήμα της περιόδου του κύματος εντοπίζεται ένα συγκεκριμένο σημείο του κύματος. Πρακτικά, ωστόσο, ένα σημαντικό στοιχείο δεν είναι τόσο η φάση ενός κύματος, όσο η μετατόπιση φάσης μεταξύ περισσότερων κυμάτων. Με άλλα λόγια, η φάση αφορά την μετατόπιση από ένα κύμα σε ένα άλλο. Η εικόνα 4α δείχνει μια κατάσταση όπου δεν υπάρχει μετατόπιση φάσης, ενώ η εικόνα 4β παρουσιάζει κύματα με μια συγκεκριμένη μετατόπιση φάσης. Η μετατόπιση φάσης μπορεί να μετρηθεί μεταξύ κάθε αντίστοιχου σημείου. Για παράδειγμα, στην εικόνα 4, η μετατόπιση φάσης απεικονίζεται ως η διαφορά μεταξύ των κορυφών των κυμάτων.



Εικ. 4 Μετατόπιση φάσης
α) καμία μετατόπιση φάσης μεταξύ των κυμάτων
β) τα κύματα έχουν μετατόπιση φάσης

1.2.6. Άλλα χαρακτηριστικά

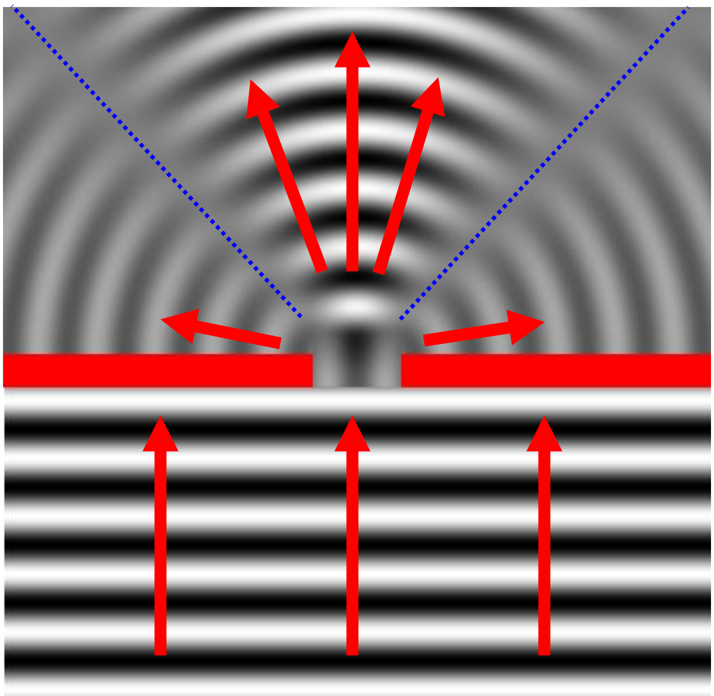
Μερικά ακόμα χαρακτηριστικά που μπορούμε να τονίσουμε είναι η ομαδική ταχύτητα κύματος και η ταχύτητα φάσης κύματος. Τα κύματα μπορεί να είναι διαμήκη ή εγκάρσια. Στην περίπτωση των εγκάρσιων μπορούμε επίσης να μιλήσουμε για πόλωση. Ωστόσο, τα χαρακτηριστικά αυτά δεν είναι απαραίτητα στην κατανόηση της ολογραφίας, και συνεπώς θα παραληφθούν στη συνέχεια του οδηγού.

1.3. Ιδιότητες των κυμάτων

Τα κύματα που μεταδίδονται στον χώρο (π.χ. ένα κύμα στο νερό, ένα ηχητικό κύμα, ένα ηλεκτρομαγνητικό κύμα) παρουσιάζουν συγκεκριμένες ιδιότητες όπως η ανάκλαση, η διάθλαση, η περίθλαση ή η συμβολή.

1.3.1. Περίθλαση

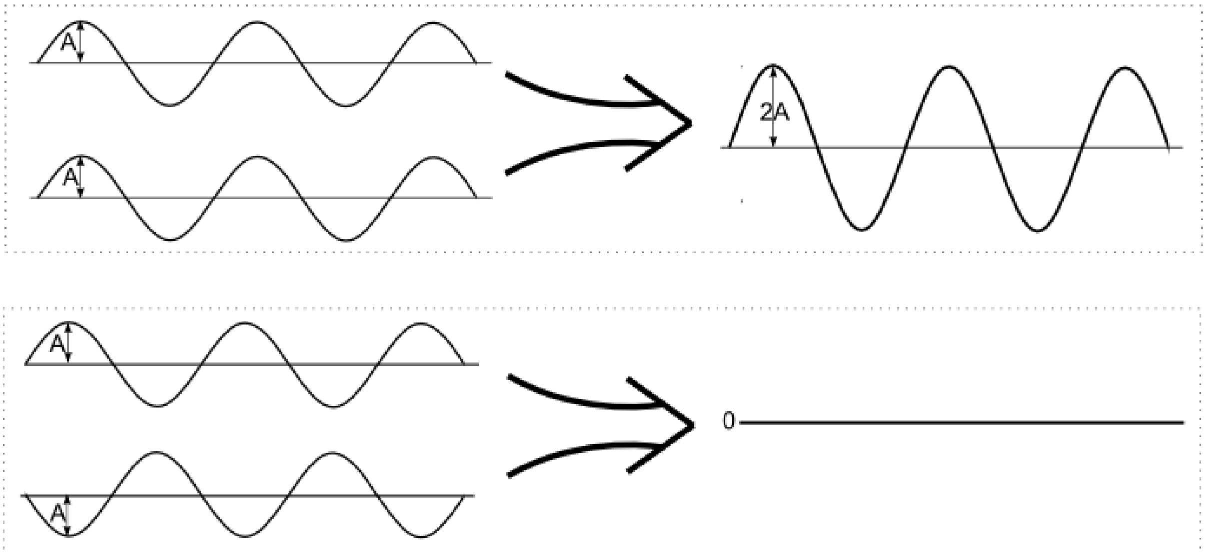
Η περίθλαση αναφέρεται στην αλλαγή της κατεύθυνσης της μετάδοσης του κύματος ως αποτέλεσμα της συνάντησής του με κάποιο εμπόδιο ή κάποια σχισμή/οπή. Αυτό το φαινόμενο γίνεται ορατό όταν το μήκος κύματος συγκρίνεται με το μέγεθος της σχισμής. Η εικόνα 5 παρουσιάζει την κατάσταση κατά την οποία το κύμα συναντά μια σχισμής. Πριν τη σχισμή, το κύμα μετακινείται σε μια κατεύθυνση κάθετη προς αυτή (η οποία σημειώνεται με κόκκινα βέλη). Υπάρχουν επίσης περιοχές απόλυτης εξαφάνισης του κύματος (σημειώνονται με μπλε γραμμές). Κατά μήκος αυτών των γραμμών το εύρος του κύματος είναι μηδέν.



Εικ. 5 Περίθλαση σε μια σχισμή

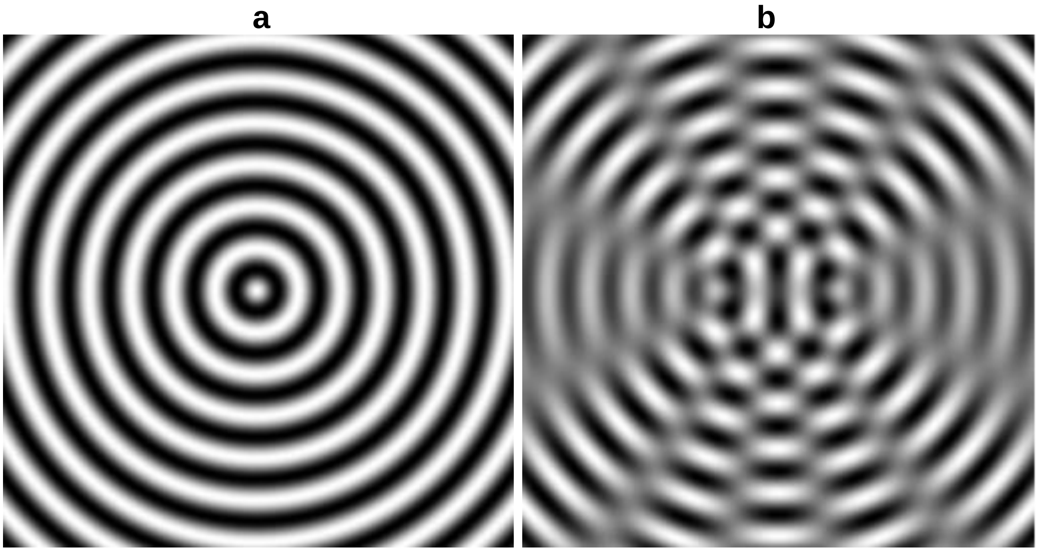
1.3.2. Συμβολή

Η συμβολή του κύματος είναι το φαινόμενο στο οποίο βασίζεται η ολογραφία. Η συμβολή είναι η αλληλοεπικάλυψη (ή η πρόσθεση) των κυμάτων. Τα μεταδιδόμενα κύματα που αλληλοεπιδρούν μεταξύ τους, μπορεί να δυναμώνουν ή να εξασθενούν. Η εικόνα 6 παρουσιάζει την ενίσχυση και την εξαφάνιση κυμάτων. Αν τα κύματα «βρίσκονται σε φάση» (η διαφορά φάσης είναι 0) τότε συμβαίνει η μεγαλύτερη δυνατή ενίσχυση κύματος. Αν όμως τα κύματα είναι «εκτός φάσης» κατά το μισό της περιόδου (η διαφορά φάσης είναι η μισή της περιόδου του κύματος), τότε τα κύματα αλληλοαναιρούνται. Όταν τα κύματα δυναμώνουν, μιλάμε για ενισχυτική συμβολή. Όταν τα κύματα εξασθενούν μιλάμε για καταστρεπτική συμβολή.



Εικ. 6 Ενισχυτική και καταστρεπτική συμβολή

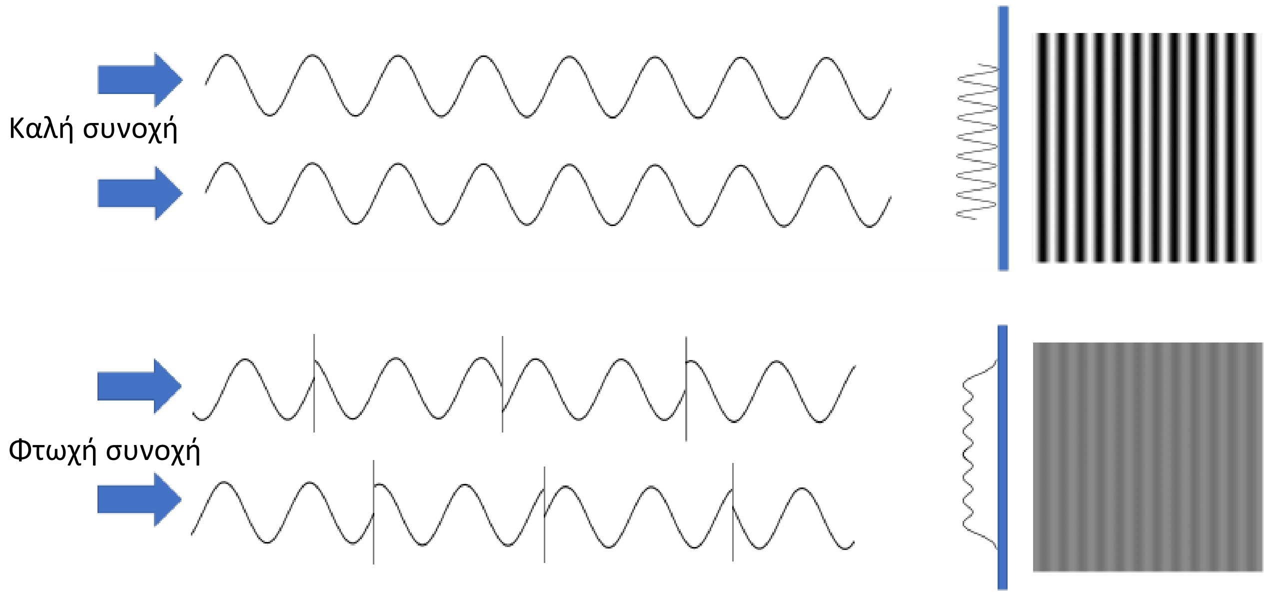
Μια καλή αναπαράσταση της συμβολής είναι αυτή των κυμάτων του νερού. Αν διεγείρουμε την επιφάνεια του νερού σε ένα σημείο, με μια συνεχή συχνότητα, τότε έχουμε ένα κύμα που μεταδίδεται ομοκεντρικά (εικόνα 7α). Σε αυτή την περίπτωση οι μαύρες περιοχές αναπαριστούν τις κοιλάδες και οι λευκές τις κορυφές. Η εικόνα 7β παρουσιάζει δύο κύματα που βρίσκονται δίπλα-δίπλα και αλληλοεπιδρούν. Οι γκρι περιοχές είναι αυτές στις οποίες το κύμα εξουδετερώνεται ενώ οι υπόλοιπες είναι αυτές που το κύμα ενισχύεται.



Εικ. 7 Το παράδειγμα των κυμάτων του νερού

1.3.1. Συνοχή

Η συνοχή των κυμάτων είναι απαραίτητη στη διατήρηση ενός συνεχούς μοτίβου συμβολής ανά τον χρόνο. Δύο κύματα είναι συνεκτικά αν έχουν μια συνεχή διαφορά φάσης. Η εικόνα 8 δείχνει – με έναν απλοποιημένο τρόπο – πώς μπορεί κάποιος να κατανοήσει τη διαφορά μεταξύ συνεκτικών και μη συνεκτικών κυμάτων. Η πάνω εικόνα δείχνει κύματα υψηλής συνοχής. Το αποτέλεσμα της αλληλοεπικάλυψης αυτών των κυμάτων είναι ο σχηματισμός σταθερών – στο χρόνο – φωτεινών και σκοτεινών κροσσών (fringes) συμβολής. Έχουμε λοιπόν τον σχηματισμό ενός μοτίβου συμβολής με καλές αντιθέσεις (οι κροσσοί είναι ευδιάκριτοι). Στην περίπτωση όμως που τα κύματα έχουν μικρό βαθμό συνοχής, οι κροσσοί που προκύπτουν σε οποιαδήποτε χρονική στιγμή έχουν κάποια μετατόπιση μεταξύ τους, και κατά συνέπεια τα μάτια μας βλέπουν ένα θολό μοτίβο συμβολής (μειωμένης αντίθεσης). Όσο πιο μικρός είναι ο βαθμός συνοχής τόσο πιο θολή είναι η εικόνα του μοτίβου. Στην περίπτωση τελείως μη συνεκτικών κυμάτων, η παρατήρηση κάποιου μοτίβου είναι αδύνατη. Στη φύση, οι περισσότερες πηγές φωτός απαρτίζονται από μη συνεκτικά κύματα. Ωστόσο, η δέσμη ενός laser είναι μια καλή πηγή συνεκτικών κυμάτων.



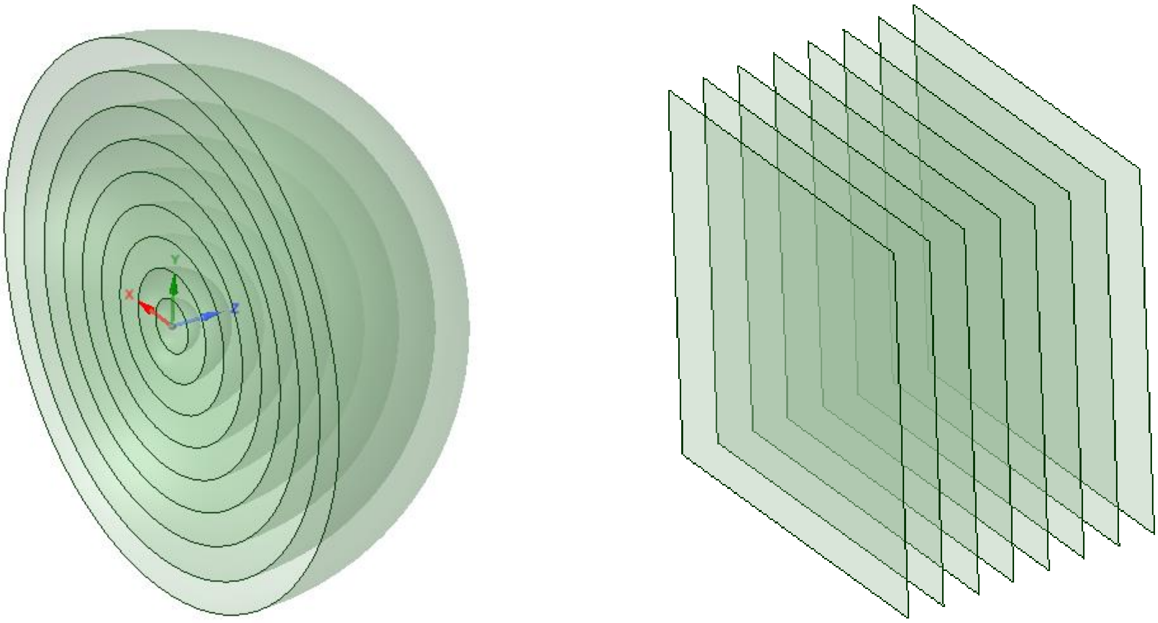
Εικ. 8 Η καλή συνοχή είναι απαραίτητη για την απόκτηση ενός σωστού μοτίβου συμβολής με καλή αντίθεση

1.4. Τα κύματα στον τρισδιάστατο χώρο

1.4.1. Σφαιρικά και επίπεδα κύματα

Στον τρισδιάστατο χώρο το κύμα μπορεί να αναπαρασταθεί ως μια επιφάνεια σε συνεχή φάση. Το πιο κοινό παράδειγμα είναι ένα σφαιρικό κύμα ή – όπως ονομάζεται – επίπεδο κύμα (εικόνα 9). Στην πρώτη περίπτωση (εικόνα 9α), οι επιφάνειες της συνεχούς φάσης (οι οποίες είναι οι επιφάνειες στις οποίες το κύμα έχει την ίδια αξία) έχουν το σχήμα σφαιρών (εξ ου και το όνομα σφαιρικό). Στη δεύτερη περίπτωση (εικόνα 9β) οι επιφάνειες σε συνεχή φάση έχουν το σχήμα επιπέδων (εξ ου και το όνομα επίπεδα κύματα).

a b

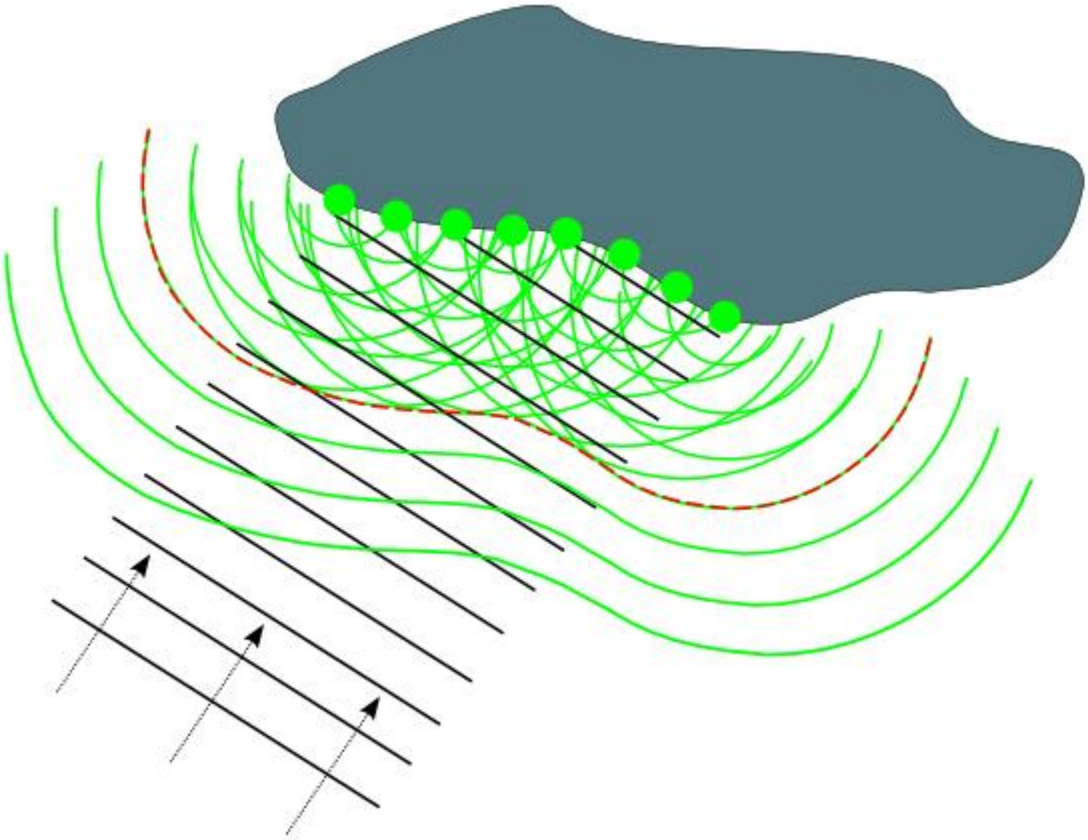


Εικ. 9 Σφαιρικά (α) και επίπεδα κύματα (β)

1.4.2. Η αρχή Huygens–Fresnel

Ας σκεφτούμε την περίπτωση όπου τα κύματα προσεγγίζουν ένα αντικείμενο (εικόνα 10). Μπορεί κανείς να υποθέσει ότι κάθε σημείο του κύματος που αγγίζει το αντικείμενο, γίνεται η πηγή σχηματισμού ενός νέου σφαιρικού κύματος. Το άθροισμα όλων αυτών των σφαιρικών κυμάτων (η συμβολή) καθορίζει το σχήμα ενός νέου κύματος. Στην εικόνα 10 το προσπίπτον κύμα σηματοδοτείται με μαύρο χρώμα. Το αντικείμενο στο οποίο φτάνει το κύμα μπορεί να αναπαρασταθεί ως μια σειρά άπειρα πολλών σημείων, τα οποία φαίνονται ως πράσινες κουκίδες στην εν λόγω εικόνα. Κάθε ένα από αυτά τα σημεία είναι η πηγή ενός δευτερεύοντος σφαιρικού κύματος. Μετά τη συνάθροιση όλων των δευτερευόντων κυμάτων, θα έχουμε ένα νέο κύμα (που στην εικόνα σηματοδοτείται με κόκκινη διακεκομμένη γραμμή) το οποίο στη συνέχεια διαδίδεται στο χώρο. Αυτή είναι η λεγόμενη αρχή των Huygens-Fresnel, η οποία κατά λέξη αποδίδεται ως:

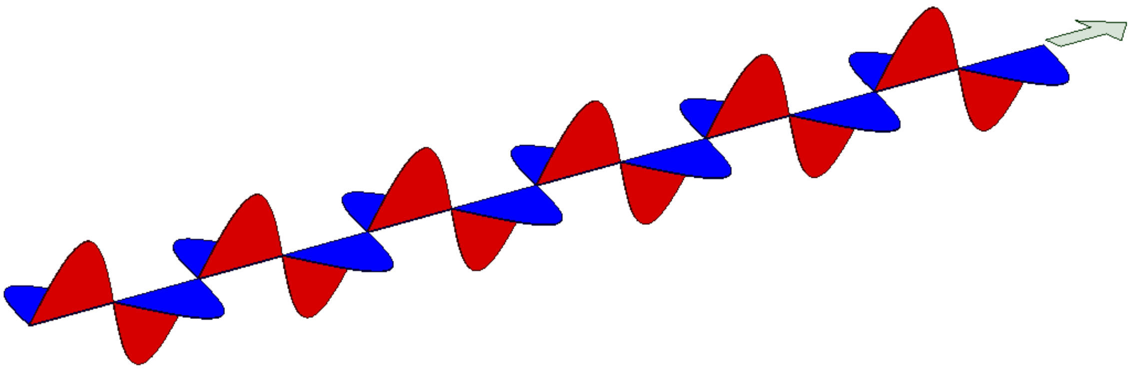
Κάθε σημείο το οποίο αγγίζεται από ένα κύμα, γίνεται η πηγή ενός σφαιρικού κύματος. Το άθροισμα αυτών των δευτερευόντων κυμάτων καθορίζει τη μορφή του κύματος σε κάθε μεταγενέστερη χρονική στιγμή.



Εικ. 10

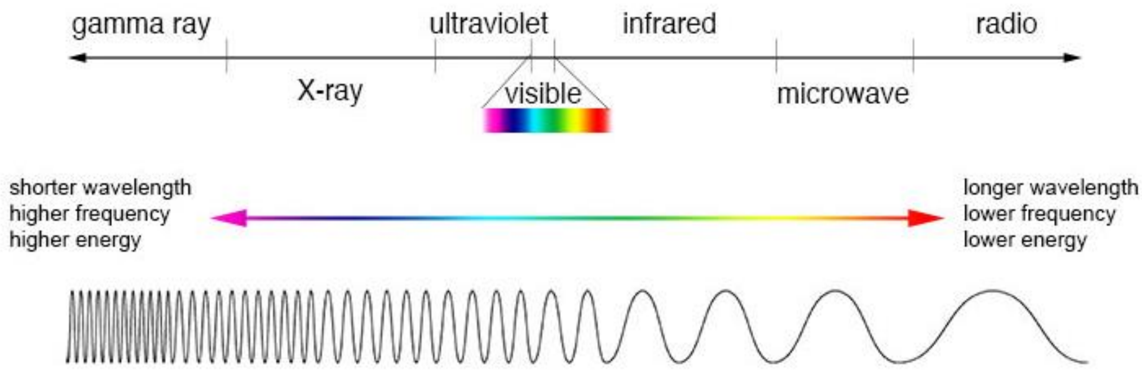
1.5. Το φως ως κύμα

Το φως είναι επίσης ένα κύμα. Αν όμως αυτό είναι αλήθεια τότε τι είδους κύμα είναι; Πρόκειται για ένα ηλεκτρομαγνητικό κύμα που προέρχεται από ένα ηλεκτρικό και ένα μαγνητικό πεδίο. Για να αναπαρασταθεί ένα τέτοιο κύμα, φανταζόμαστε δύο αυξομειούμενα, και κάθετα μεταξύ τους, πεδία που διαδίδονται στο χώρο (εικόνα 11). Το ηλεκτρικό πεδίο σηματοδοτείται με κόκκινο και το μαγνητικό με μπλε χρώμα.



Εικ. 11 Ηλεκτρομαγνητικό κύμα

Στη φύση υπάρχουν πολλών ειδών ηλεκτρομαγνητικές ακτινοβολίες οι οποίες κατηγοριοποιούνται ανάλογα με την ταχύτητα αλλαγής του ηλεκτρομαγνητικού πεδίου στο χώρο (δηλαδή ανάλογα της συχνότητας των κυμάτων). Τα ηλεκτρομαγνητικά κύματα με την υψηλότερη συχνότητα είναι οι ακτίνες γάμα. Ακολουθούν οι ακτίνες Χ, οι υπεριώδεις, το ορατό φως, οι υπέρυθρες, τα μικροκύματα και τα ραδιοκύματα, τα οποία και έχουν τη χαμηλότερη ηλεκτρομαγνητική συχνότητα (εικόνα 12).

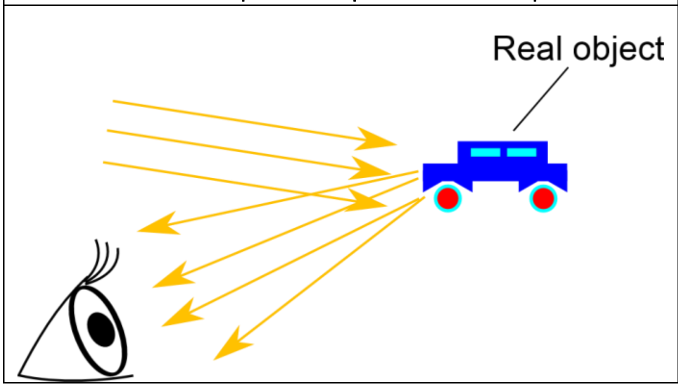
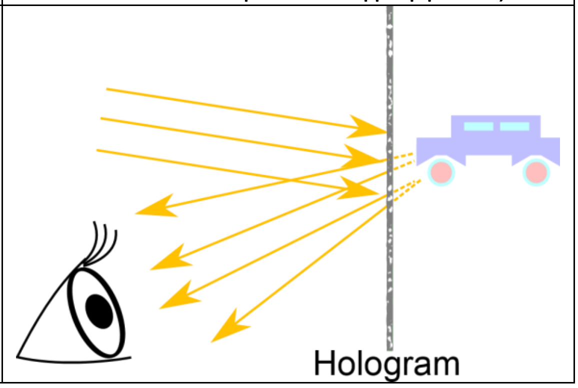


Εικ. 12 Το φάσμα των ηλεκτρομαγνητικών κυμάτων. Πηγή: NASA's Imagine the Universe (https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html)

Κεφάλαιο 2.

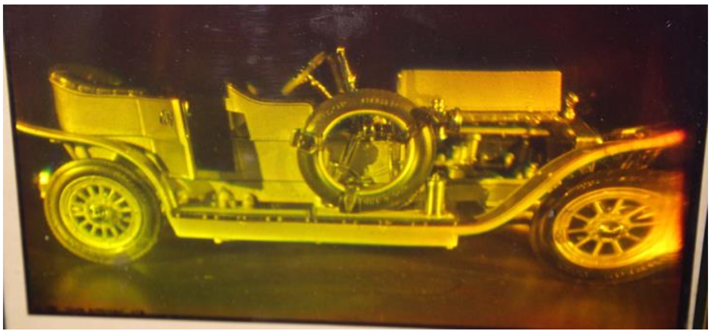
2.1. Η Ολογραφία με απλά λόγια

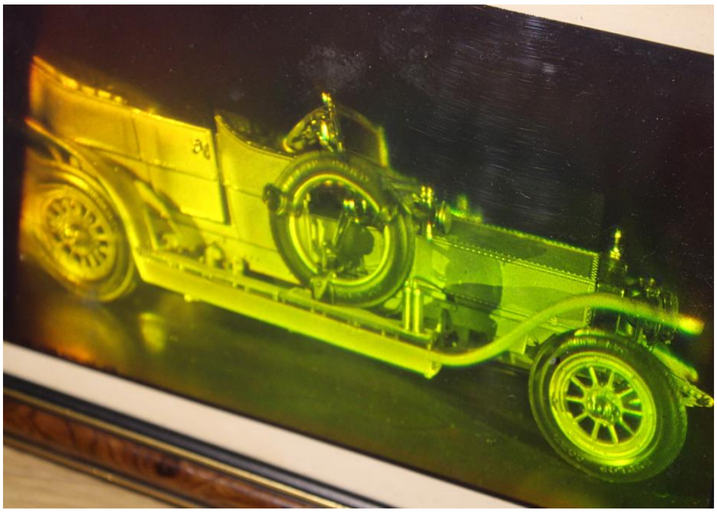
Το ολόγραμμα είναι μια επιφάνεια ή ένα γυάλινο πλακίδιο στην επιφάνεια ή στον όγκο του οποίου καταγράφεται ένα μοτίβο συμβολής. (interference pattern). Το φως που περνά ή που ανακλάται από το ολόγραμμα, δημιουργεί μια όμοια εικόνα με αυτή του φωτός που αντανακλάται από το αληθινό αντικείμενο (εικόνα 13).

Αντανάκλαση του αληθινού αντικειμένου	Αντανάκλαση του ολογράμματος
	

Εικ. 13 Το φως που αντανακλάται από το ολόγραμμα συμπεριφέρεται με τον ίδιο τρόπο που θα έκανε αν ερχόταν από το αληθινό αντικείμενο

Ένα χαρακτηριστικό γνώρισμα του ολογράμματος είναι ότι το ολογραφημένο αντικείμενο είναι πλήρως τρισδιάστατο. Αυτό σημαίνει ότι όταν κάποιος κοιτά το ολόγραμμα από μια συγκεκριμένη γωνία, μπορεί να δει λεπτομέρειες που δεν είναι ορατές όταν κοιτάξει το ίδιο ολόγραμμα από άλλη οπτική γωνία (εικόνα 14). Αυτού του είδους η ιδιότητα δε χαρακτηρίζει τις φωτογραφίες, καθώς οι φωτογραφίες φαίνονται ίδιες από όποια γωνία και αν τις κοιτάξει κανείς.

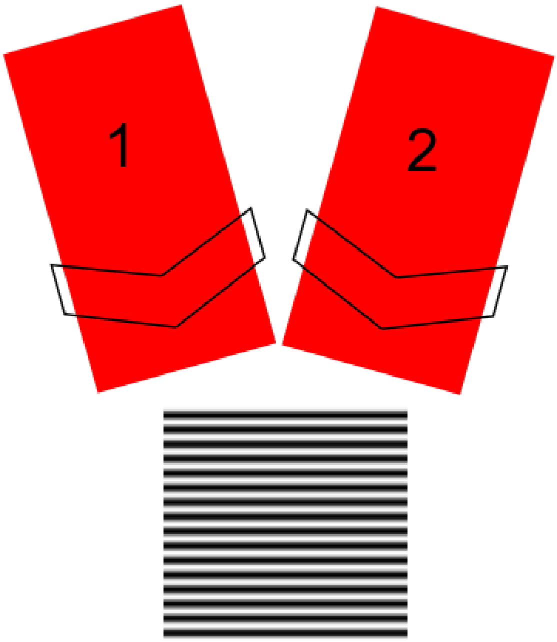




Εικ. 14 Το ολόγραμμα αλλάζει ανάλογα με τη γωνία παρατήρησης

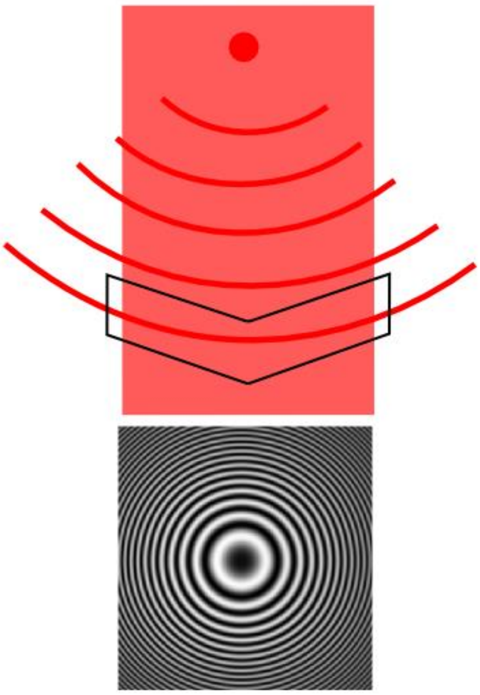
2.2. Η αρχή του σχηματισμού ολογράμματος

Το φυσικό φαινόμενο που δημιουργεί ένα ολόγραμμα είναι αυτό της συμβολής. Έχουμε δει ήδη ότι δύο συνεκτικά κύματα συμβάλουν το ένα του άλλου ώστε να δημιουργηθούν κροσσοί συμβολής. Η εικόνα 15 δείχνει ένα τέτοιο παράδειγμα μέσω δύο επίπεδων κυμάτων που συμβάλλουν έτσι ώστε να δημιουργηθούν ίσιοι κροσσοί.



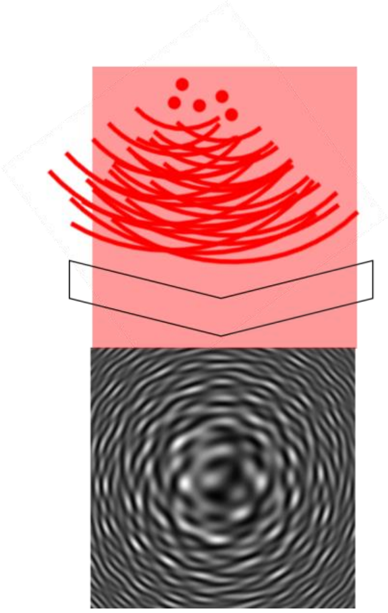
Εικ. 15 Συμβολή δύο επίπεδων κυμάτων

Η συμβολή ενός επίπεδου κύματος με μια σημειακή πηγή, απεικονίζεται στην εικόνα 16. Οι κροσσοί συμβολής έχουν κυκλικό σχήμα. Μπορούμε να πούμε λοιπόν ότι ένα τέτοιο μοτίβο συμβολής περιέχει πληροφορίες για ένα σημείο στο χώρο. Αν καταφέρουμε να καταγράψουμε αυτή τη διαταραχή, σε ένα ολογραφικό φιλμ για παράδειγμα, τότε θα είναι πιθανό να μπορέσουμε να ανακατασκευάσουμε το αποκωδικοποιημένο σημείο με το να φωτίσουμε το μοτίβο με ένα επίπεδο κύμα.



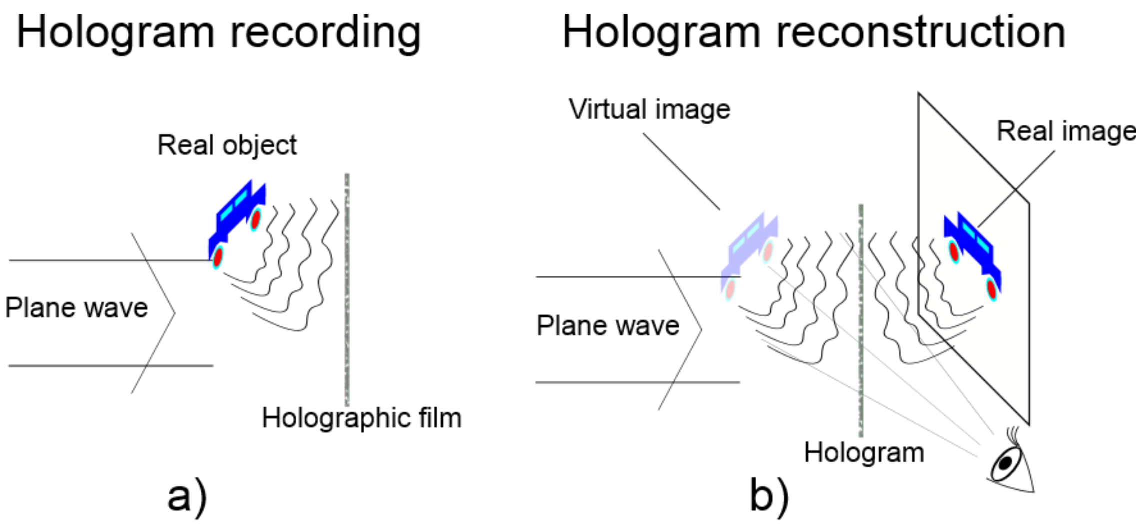
Εικ. 16 Η συμβολή ενός επίπεδου κύματος με μια σημειακή πηγή

Κάθε αντικείμενο που φωτίζεται μπορεί να αντιμετωπιστεί ως ένα σετ από πολλές δευτερεύουσες σημειακές πηγές (αρχή Huygens-Fresnel). Το μοτίβο συμβολής που προκύπτει από τη συμβολή ενός επίπεδου κύματος, με πολλές σημειακές πηγές, θεωρείται γενικά κάτι πολύπλοκο ως προς την κατανόησή του (βλέπε εικόνα 17). Ως γενικό κανόνα μπορούμε να πούμε ότι το αντικείμενο κωδικοποιείται σε ένα μοτίβο συμβολής.



Εικ. 17 Η συμβολή ενός επίπεδου κύματος με πολλές σημειακές πηγές

Ένα τέτοιο μοτίβο είναι το ολόγραμμα. Το ολόγραμμα μπορεί να ανακατασκευαστεί όταν φωτιστεί με ένα επίπεδο κύμα. Η εικόνα 18 παρουσιάζει ένα παράδειγμα της καταγραφής και της ανακατασκευής ενός ολογράμματος. Κατά τη διάρκεια της ανακατασκευής, αποκτούμε μια αληθινή εικόνα (ορατή στην οθόνη) και μια εικονική (ορατή όταν κοιτάμε με άμεσο τρόπο το ολόγραμμα που φωτίζεται).



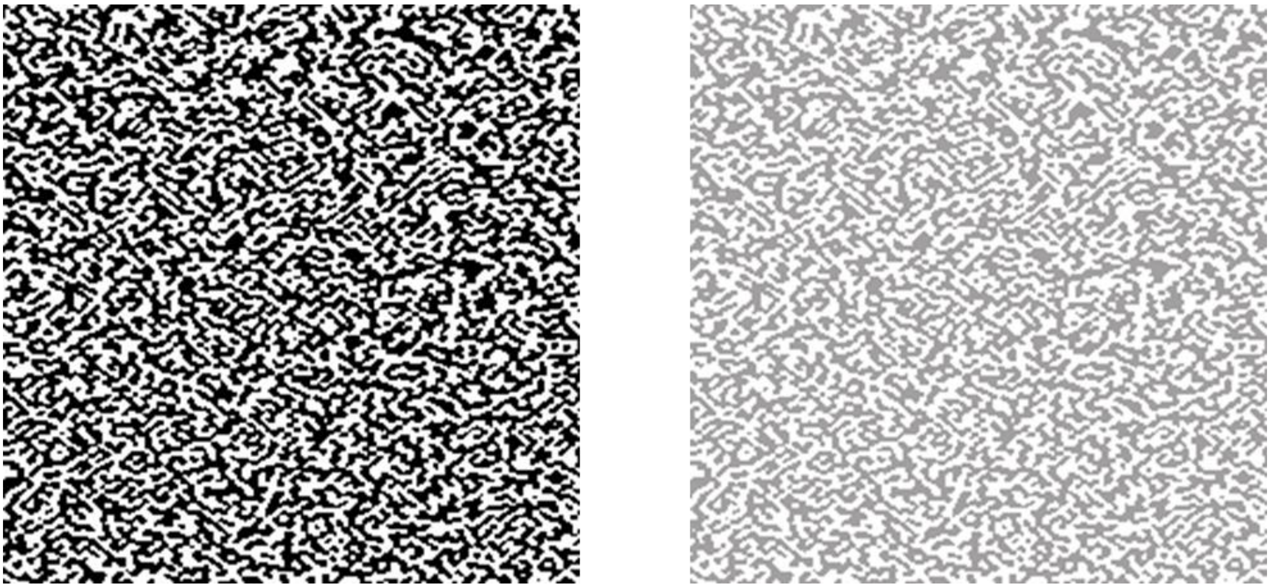
Εικ. 18 Καταγράφοντας (a) και ανακατασκευάζοντας (b) ένα ολόγραμμα

2.3. Τύποι και ιδιότητες των ολογραμμάτων

2.3.1. Τύποι ολογραμμάτων

Τα ολογράμματα μπορούν να ταξινομηθούν με διαφορετικούς τρόπους, ανάλογα με τις ιδιότητές τους. Ένας τρόπος αφορά τον τρόπο με τον οποίο το φως περνά μέσω του ολογράμματος. Τότε λέμε ότι το ολόγραμμα μπορεί να είναι διαμορφωμένο βάσει πλάτους ή βάσει φάσης. Η διαμόρφωση βάσει πλάτους σημαίνει ότι το μοτίβο συμβολής είναι εγγεγραμμένο με τη μορφή περισσότερο ή λιγότερο διάφανων περιοχών (εικόνα 19α). Η διαμόρφωση βάσει φάσης είναι εντελώς διάφανη και το μοτίβο συμβολής εγγράφεται με τη μορφή περιοχών με διαφορετικό δείκτη διάθλασης (εικόνα 19β).

a b



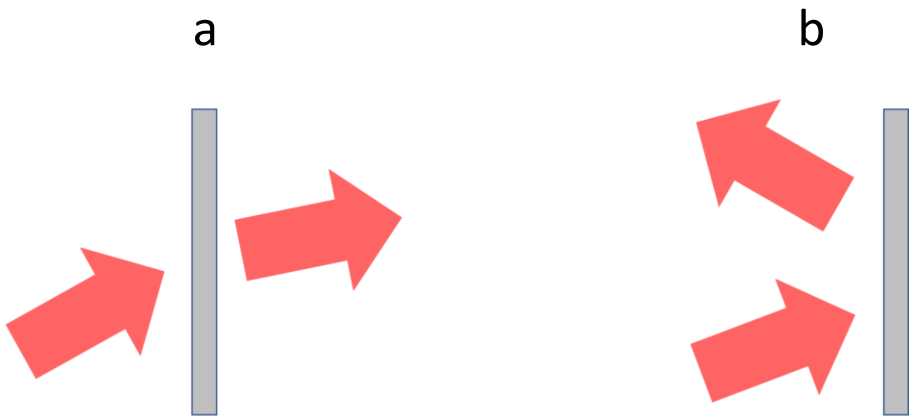
Εικ. 19 Ολογράμματα διαμορφωμένα βάσει πλάτους (a) και φάσης (b)

Τα ολογράμματα μπορεί επίσης να διακρίνονται σε λεπτά και πυκνά (εικόνα 20). Τα λεπτά ολογράμματα είναι αυτά που συμπεριφέρονται σαν το ολόγραμμα να έχει εγγραφεί μόνο στην επιφάνεια του ολογραφικού πλακιδίου. Τα πυκνά είναι αυτά στα οποία το ολόγραμμα εγγράφεται σε όλο τον όγκο του ολογραφικού πλακιδίου.

a b

Εικ. 20 Λεπτό (a) και πυκνό (b) ολόγραμμα

Άλλος τρόπος διάκρισης είναι σε ολογράμματα μετάδοσης και ανάκλασης. Τα ολογράμματα μετάδοσης μπορούν να παρατηρηθούν με τη μετάδοση φωτός μέσω ενός ολογράμματος, ενώ τα ολογράμματα ανάκλασης μπορούν να παρατηρηθούν με την ανάκλαση του φωτός από ένα ολόγραμμα.



Εικ. 21 Ολογράμματα μετάδοσης (a) και ανάκλασης (b)

2.3.1. Ιδιότητες των ολογραμμάτων

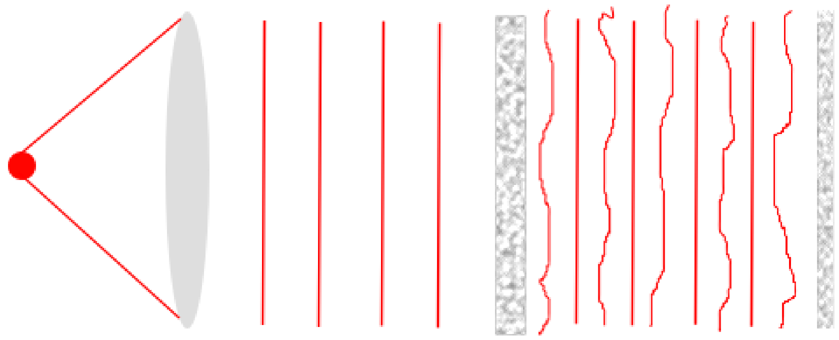
Η ολογραφία είναι μια φωτογραφική μέθοδος-πληροφορία της εικόνα που εγγράφεται στην ολογραφική επιφάνεια. Ωστόσο, οι διαφορές μεταξύ των ολογραμμάτων και των φωτογραφιών είναι θεμελιώδεις:

- Στην περίπτωση της ολογραφίας, το φιλμ πρέπει να έχει πολύ μεγαλύτερη ποιότητα-ανάλυση (μετριέται σε γραμμές ανά χιλιοστόμετρο).
- Το φωτογραφικό αρνητικό περιέχει πληροφορίες μόνο για ότι αφορά την ένταση του φωτός, ενώ στο ολόγραμμα (με τη μορφή ενός μοτίβου συμβολής), εγγράφεται τόσο η πληροφορία για την ένταση όσο και για τη φάση του φωτός (δηλαδή την απόσταση των σημείων του αντικειμένων από το ολόγραμμα).
- Κάθε απόσπασμα του φωτογραφικού αρνητικού περιέχει πληροφορίες για τα διαφορετικά μέρη της εικόνα, ενώ στην περίπτωση του ολογράμματος, κάθε κομμάτι περιέχει πληροφορίες για ολόκληρη την εικόνα. Αυτό σημαίνει ότι αν κόψουμε το ολόγραμμα σε δύο μέρη, κάθε ένα από αυτά θα ανακατασκευάσει μια ολόκληρη εικόνα. Βέβαια, θα υπάρχει μια διαφορά στη φωτεινότητα της εικόνας, ενώ η γωνία παρατήρησης του ολογράμματος θα μικρύνει.

2.4. Βασικές διατάξεις της ολογραφικής εγγραφής

2.4.1. Το ολόγραμμα Gabor

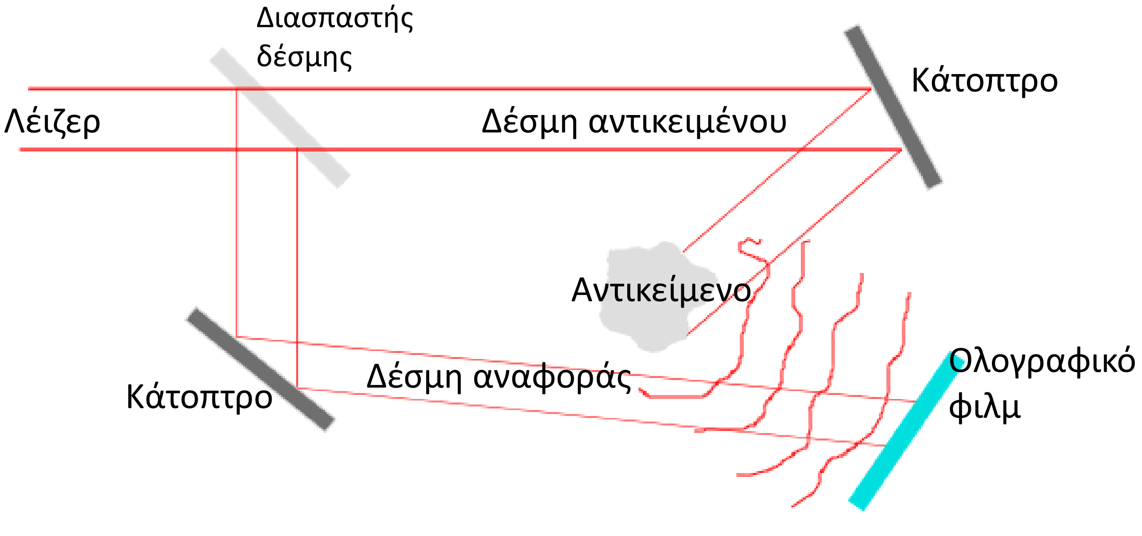
Ιστορικά, η πρώτη διάταξη για την εγγραφή ολογραμμάτων προτάθηκε από τον Denis Gabor το 1948. Οι πηγές λέιζερ δεν είχαν ακόμα ανακαλυφθεί. Έτσι η συγκεκριμένη διάταξη χρησιμοποιούσε μια σημειακή πηγή φωτός, η οποία ήταν μερικώς συνεκτική και συνεπώς ικανή να εγγράψει ολόγραμμα. Το αντικείμενο ήταν ένα ημιδιαφανές φιλμ (π.χ. ένα σύμβολο ή μια απλή εικόνα). Μέρος του φωτός περνάει άμεσα το φιλμ, και μέρος του σκεδάζεται. Και οι δυο ακτίνες συμβάλλουν η μία της άλλης, με το να εγγράφονται στο φιλμ ως μοτίβο συμβολής (εικόνα 22).



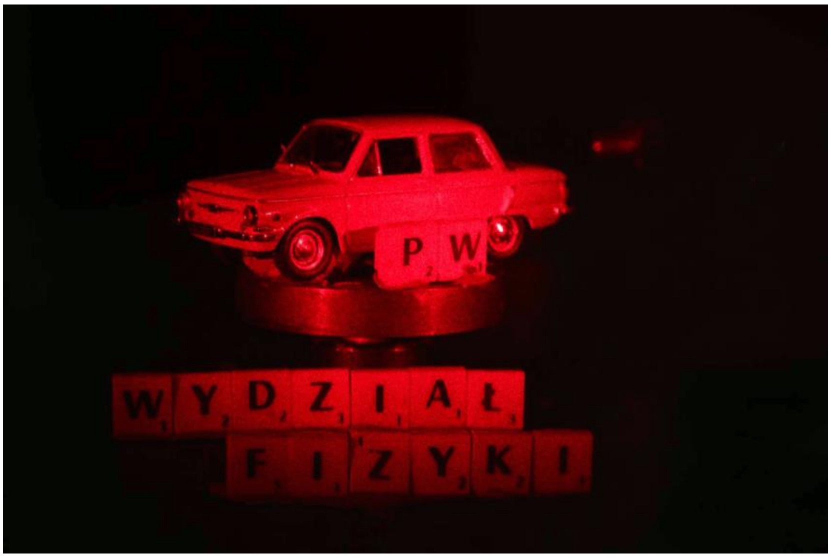
Εικ. 22 Ολόγραμμα Gabor

2.4.1. Διάταξη Leith-Upatnieks

Σε αυτή τη διάταξη, κάποιος μπορεί να εγγράψει τα λεγόμενα ολογράμματα Fresnel, τα οποία μπορούν να ανακατασκευαστούν με το φως του λέιζερ. Η δέσμη του λέιζερ διαιρείται σε δύο (στη δέσμη αναφοράς και στη δέσμη αντικειμένου). Το φως σκεδάζεται από το αντικείμενο, συμβάλει με τη δέσμη αναφοράς, και εγγράφεται ως ένα μοτίβο συμβολής στο ολογραφικό φιλμ (εικόνα 23). Ένα παράδειγμα ολογράμματος Fresnel δίνεται στην εικόνα 24.



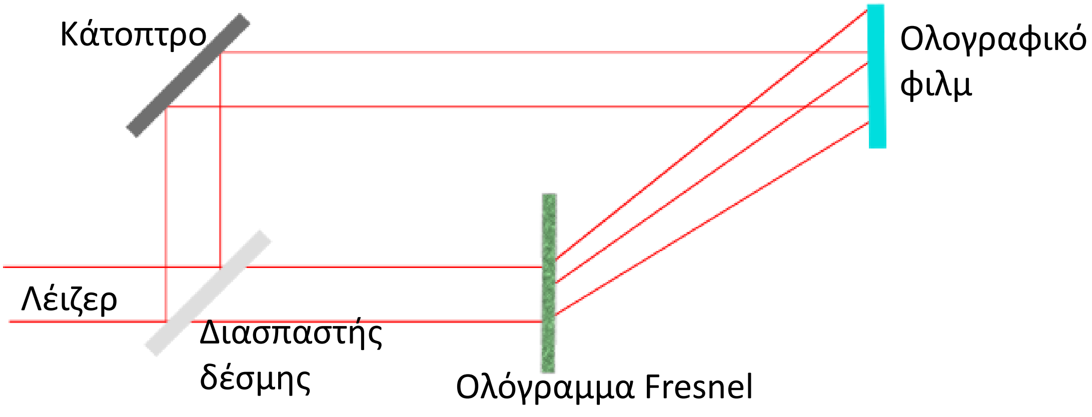
Εικ. 23 Καταγραφή ολογράμματος Fresnel με τη διάταξη Leith-Upatnieks



Εικ. 24 An example of Fresnel hologram

2.4.1. Ολόγραμμα Rainbow (Benton)

Σε αυτή τη διάταξη (εικόνα 25) το αντικείμενο ολογράφησης είναι ένα ολόγραμμα Fresnel. Το ολόγραμμα που καταγράφεται έχει το πλεονέκτημα ότι μπορεί να είναι ορατό στο λευκό φως. Γυρνώντας το ολόγραμμα ελαφρώς προς μία κατεύθυνση, καθίσταται δυνατή η ορατότητα του τρισδιάστατου εφέ (διαδικασία που ονομάζεται παράλλαξη), ενώ στην άλλη κατεύθυνση είναι δυνατή η παρατήρηση αλλαγής χρωμάτων. Ένα τέτοιο παράδειγμα παρουσιάζεται στην εικόνα 26.

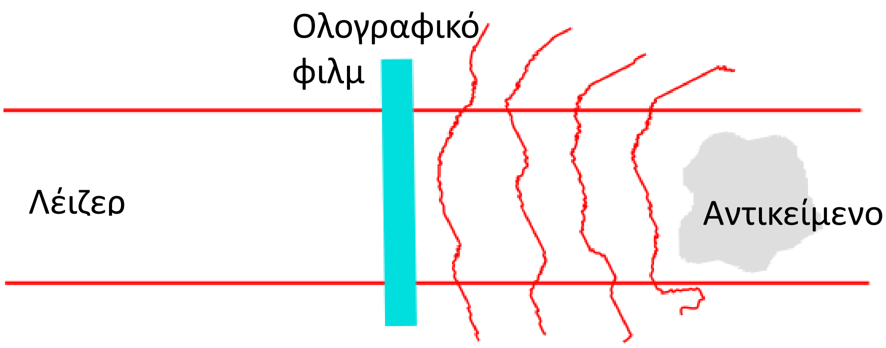


Εικ. 25 Καταγράφοντας ένα ολόγραμμα rainbow

Εικ. 26 Παράδειγμα ενός ολογράμματος rainbow

2.4.1. Ολογράμματα όγκου

Ένα τέτοιο ολόγραμμα μπορεί να καταγραφεί με τη διάταξη της εικόνας 27. Το ολόγραμμα όγκου μπορεί να γίνει ορατό και στην κατάσταση μετάδοσης (transmission), με λευκό φως. Ένα παράδειγμα τέτοιου ολογράμματος δίνεται στην εικόνα 28..



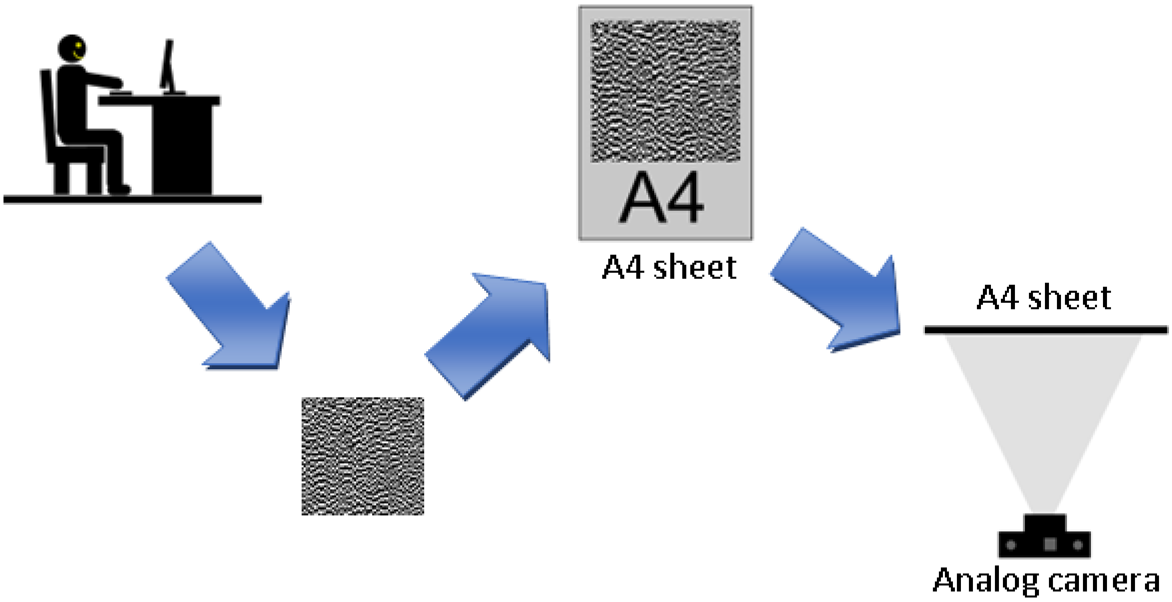
Εικ. 27 Διάταξη για την εγγραφή ολογράμματος όγκου

Εικ. 28 Παράδειγμα ολογράμματος όγκου

2.4.1. Ολόγραμμα παραγόμενο μέσω υπολογιστή (CGH)

Σε αυτή την περίπτωση ολογράμματος δεν απαιτείται κάποιο φυσικό αντικείμενο. Στην απλούστερη μορφή, το ρόλο του αντικειμένου παίζει ένα γραφικό αρχείο (graphic file) που περιέχει κάποιο σύμβολο ή ένα δισδιάστατο σχήμα. Στη βάση ενός τέτοιου αρχείου, υπάρχει ένα μοτίβο συμβολής (επίσης με τη μορφή ενός γραφικού αρχείου) του οποίου η επεξεργασία γίνεται μέσω του υπολογιστή. Για τους υπολογισμούς γίνεται χρήση ενός επαναληπτικού αλγορίθμου (ο πιο διαδεδομένος είναι ο αλγόριθμος Gerchberg Saxton). Το υπολογιζόμενο μοτίβο συμβολής μπορεί να δημιουργηθεί με τη μορφή ενός οπτικού αντικειμένου (μιας διαφάνειας για παράδειγμα). Αν φωτίσουμε τη διαφάνεια με μια δέσμη λέιζερ, τότε το μεταδιδόμενο φως θα σχηματίζει το σύμβολο ή το δισδιάστατο σχήμα, από το οποίο προήλθε το μοτίβο.

Η δημιουργία ενός ολογράμματος υψηλής ποιότητας απαιτεί πολύπλοκο εξοπλισμό. Ωστόσο, ένα ολόγραμμα μέσω υπολογιστή μπορεί πιο εύκολα να παραχθεί, γεγονός που καθιστά πιο εύκολη τη χρήση του για εκπαιδευτικούς σκοπούς. Σε αυτή την περίπτωση, το υπολογισμένο μοτίβο συμβολής θα πρέπει να εκτυπωθεί σε ένα απλό χαρτί (π.χ. μια Α4), και να ληφθεί μια φωτογραφία του με αναλογική μηχανή. Στη συνέχεια, θα πρέπει να γίνει εμφάνιση του φιλμ (σε σκοτεινό θάλαμο) ώστε να μπορέσει να αναπαραχθεί το ολόγραμμα. Η διαδικασία αυτή παρουσιάζεται και στην εικόνα 29.

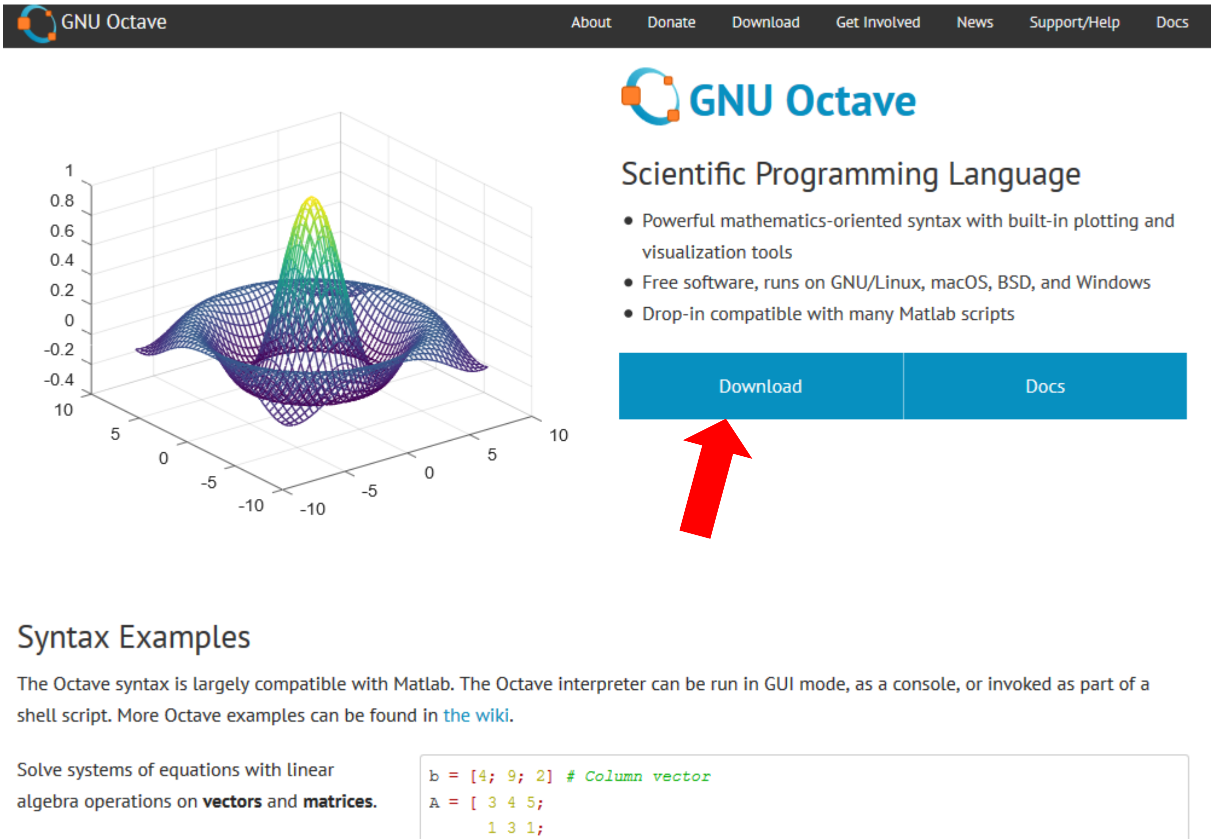


Εικ. 29 Η διαδικασία απόκτησης ενός ολογράμματος παραγόμενου μέσω υπολογιστή

Κεφάλαιο 3. Επεξεργασία εικόνα με τη χρήση του λογισμικού Octave

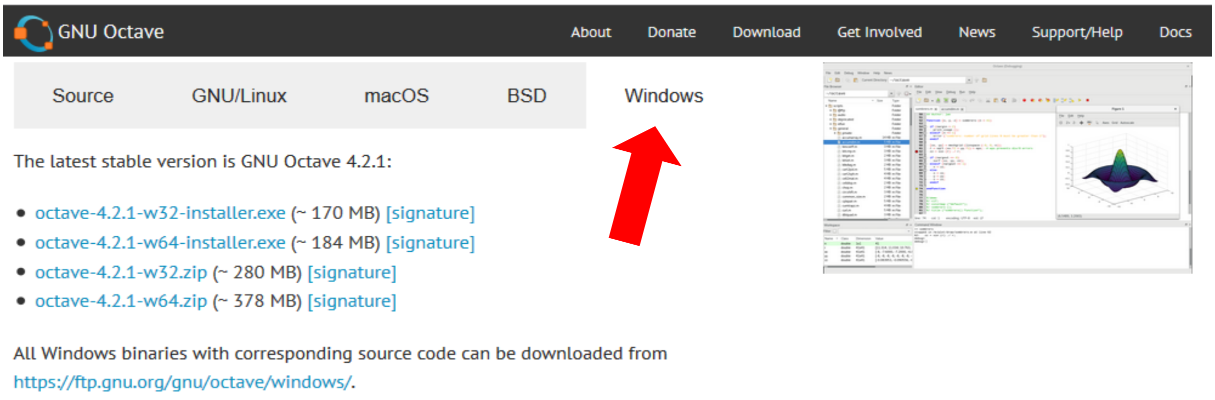
3.1. Εγκαθιστώντας το Octave

Το λογισμικό Octave θα πρέπει να ληφθεί από την ιστοσελίδα:
https://www.gnu.org/software/octave/
Για να γίνει αυτό, θα πρέπει να επιλέξετε download, όπως φαίνεται και στην εικόνα 30:



Εικ. 30 Κατεβάζοντας το Octave

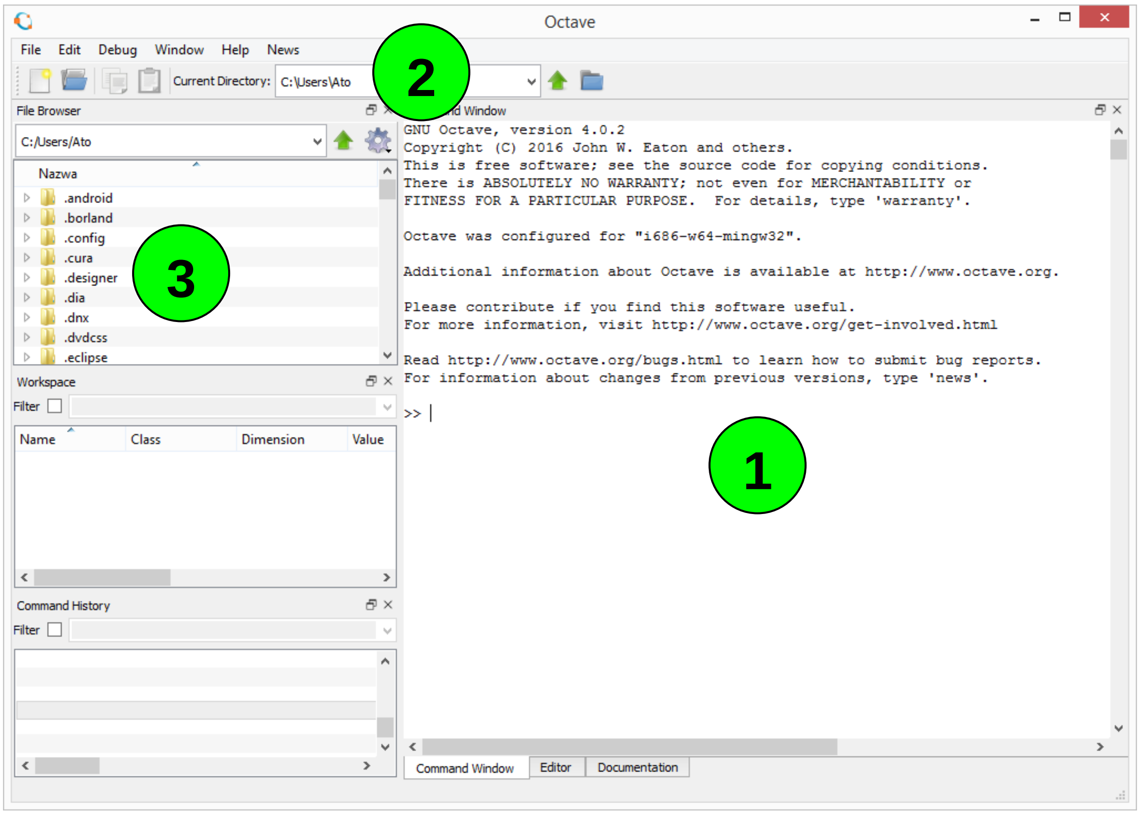
Μετά, πηγαίνετε στο μενού Windows και κατεβάζετε το αρχείο octave-4.2.1-w32-installer.exe ή octave-4.2.1-w64-installer.exe ανάλογα με την έκδοση που διαθέτετε (32 bit ή 64 bit, Εικόνα 31).



Εικ. 31 Κατεβάζοντας το Octave

3.2. Εκκίνηση του προγράμματος

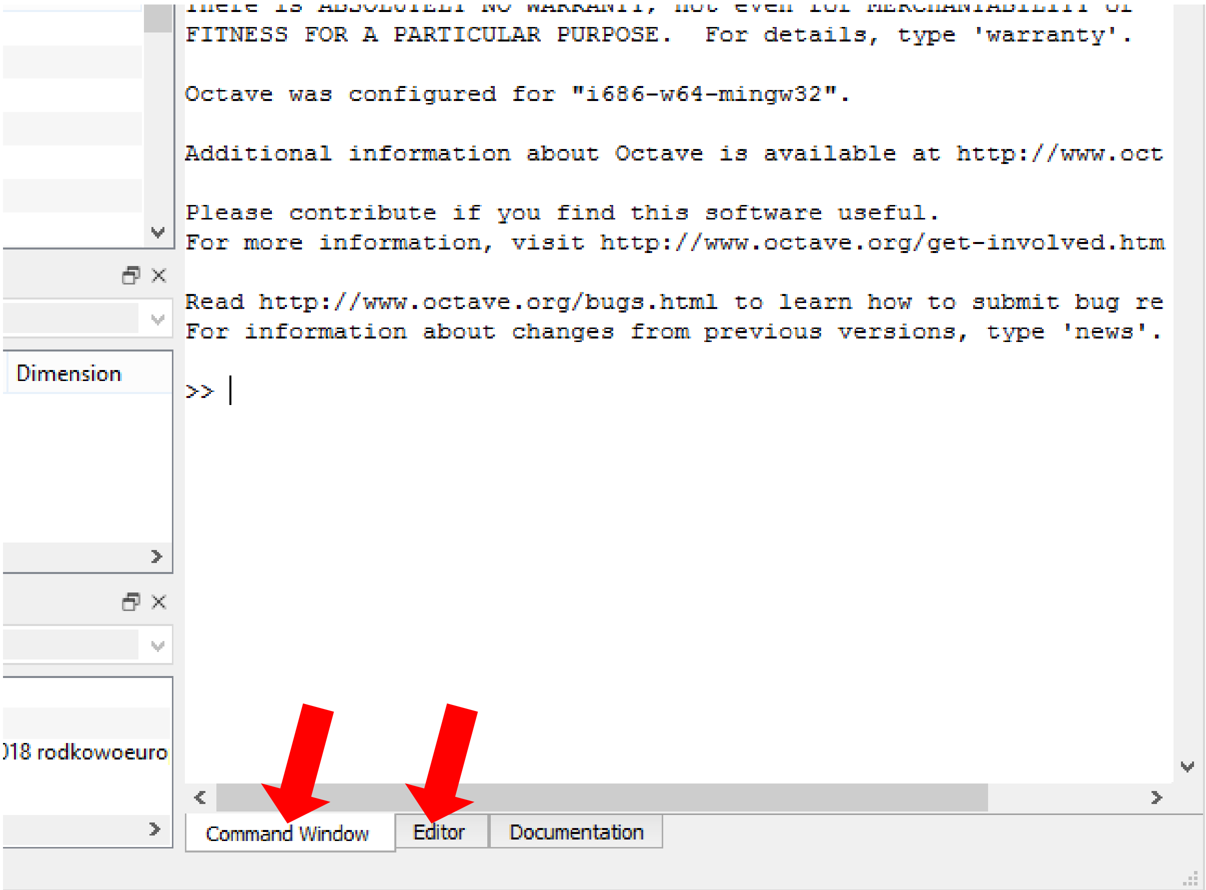
Αν δουλεύεται σε περιβάλλον Windows, κάντε κλικ στο μενού Start και γράψτε “Octave” στο πεδίο αναζήτησης. Μετά ανοίξτε το αρχείο “Octave (GUI)”. Το κυρίως παράθυρο θα εμφανιστεί. Σε αυτό υπάρχουν αρκετά επιμέρους τμήματα (Εικόνα 32). Το πιο σημαντικό από αυτά είναι η περιοχή όπου μπορείτε να εισάγεται εντολές (σημειωμένο με τον αριθμό 1). Υπάρχει επίσης το παράθυρο που μας επιτρέπει να επιλέξουμε τον φάκελο στον οποίο θα δουλεύουμε (νούμερο 2 στην εικόνα) και τέλος υπάρχει ένα μενού περιήγησης αρχείων του τρέχοντος φακέλου (νούμερο 3).



Εικ. 32 Εκκινώντας το Octave

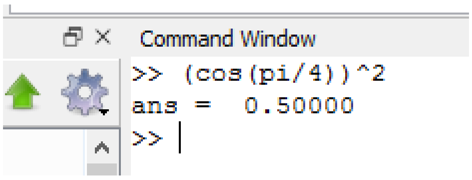
3.3. Παράθυρο εντολών (Command Window)

Η εισαγωγή εντολών μπορεί να γίνει με δύο τρόπους: με τη χρήση του "Command Window", ή μέσω του "Editor". Στην πρώτη περίπτωση, κάθε εντολή εκτελείται άμεσα, από τη στιγμή που θα πατηθεί το Enter. Στη δεύτερη περίπτωση, ο λεγόμενος κώδικας θα πρέπει να γραφτεί, και θα τρέξει μόνο τη στιγμή που θα επιλεγεί η εκτέλεση αυτού. Στη συνέχεια παρατίθενται ορισμένα παραδείγματα, για την καλύτερη κατανόηση των παραπάνω. Ο τρόπος μετάβασης από το command window στο editor φαίνεται στην εικόνα 33.



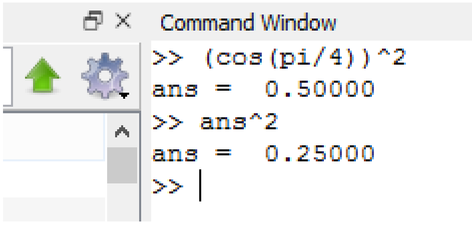
Εικ. 33 Το Command Window και το Editor

Το Command Window χρησιμοποιείται για την εκτέλεση σύντομων και γρήγορων υπολογισμών. Για παράδειγμα, αν θέλουμε να υπολογίσουμε την αξία του $\cos^2\left(\frac{\pi}{4}\right)$, τότε στο παράθυρο πρέπει απλά να εισάγουμε την πρόταση `(cos(pi/4))^2`. Ο χειριστής ^ σημαίνει την ύψωση στη δύναμη (στην περίπτωσή μας τη δύναμη του 2). Όταν εισάγουμε το παραπάνω, και το επιβεβαιώσουμε πατώντας το Enter, θα λάβουμε το αποτέλεσμα που φαίνεται στην εικόνα 34.



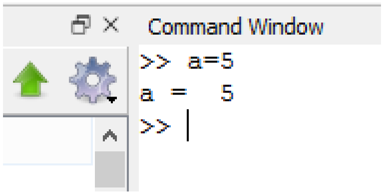
Εικ. 34 Αποτέλεσμα υπολογισμού

Το αποτέλεσμα της πράξης εκφράζεται αυτόματα μέσω της μεταβλητής "ans". Μπορούμε να χρησιμοποιήσουμε αυτή τη μεταβλητή για περαιτέρω υπολογισμούς με έναν πολύ απλό τρόπο. Για παράδειγμα, αν θέλουμε να έχουμε ξανά το αποτέλεσμα, υψωμένο στη δύναμη 2, θα πρέπει απλά να πληκτρολογήσουμε `ans ^ 2`, όπως φαίνεται και στην εικόνα 35. Η μεταβλητή "ans" θα πάρει μια νέα τιμή, που στην περίπτωσή μας ισούται με 0.25.



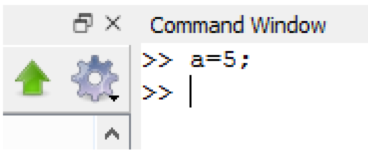
Εικ. 35 Περαιτέρω υπολογισμοί

Φυσικά, μπορούμε να αποθηκεύσουμε διαφορετικές τιμές σε ανεξάρτητες μεταβλητές. Αν για παράδειγμα, θέλουμε να θυμόμαστε μια τιμή αξίας 5, θα ονομάσουμε τη μεταβλητή μας "a" (για παράδειγμα), και απλά θα γράψουμε a = 5 (εικόνα 36).



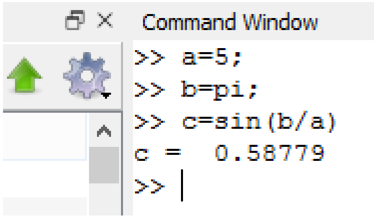
Εικ. 36 Δήλωση της μεταβλητής

Με αυτό τον τρόπο δηλώνουμε μια νέα μεταβλητή που ονομάζεται "a" και έχει τιμή 5. Αφού πατήσουμε το Enter, το αποτέλεσμα της εντολής μας αναγράφεται στην οθόνη (`a = 5`). Αν δε θέλουμε να φαίνονται τα αποτελέσματα στην οθόνη θα πρέπει στο τέλος της εντολής να εισάγουμε το ελληνικό ερωτηματικό (semi-colon) (εικόνα 37).



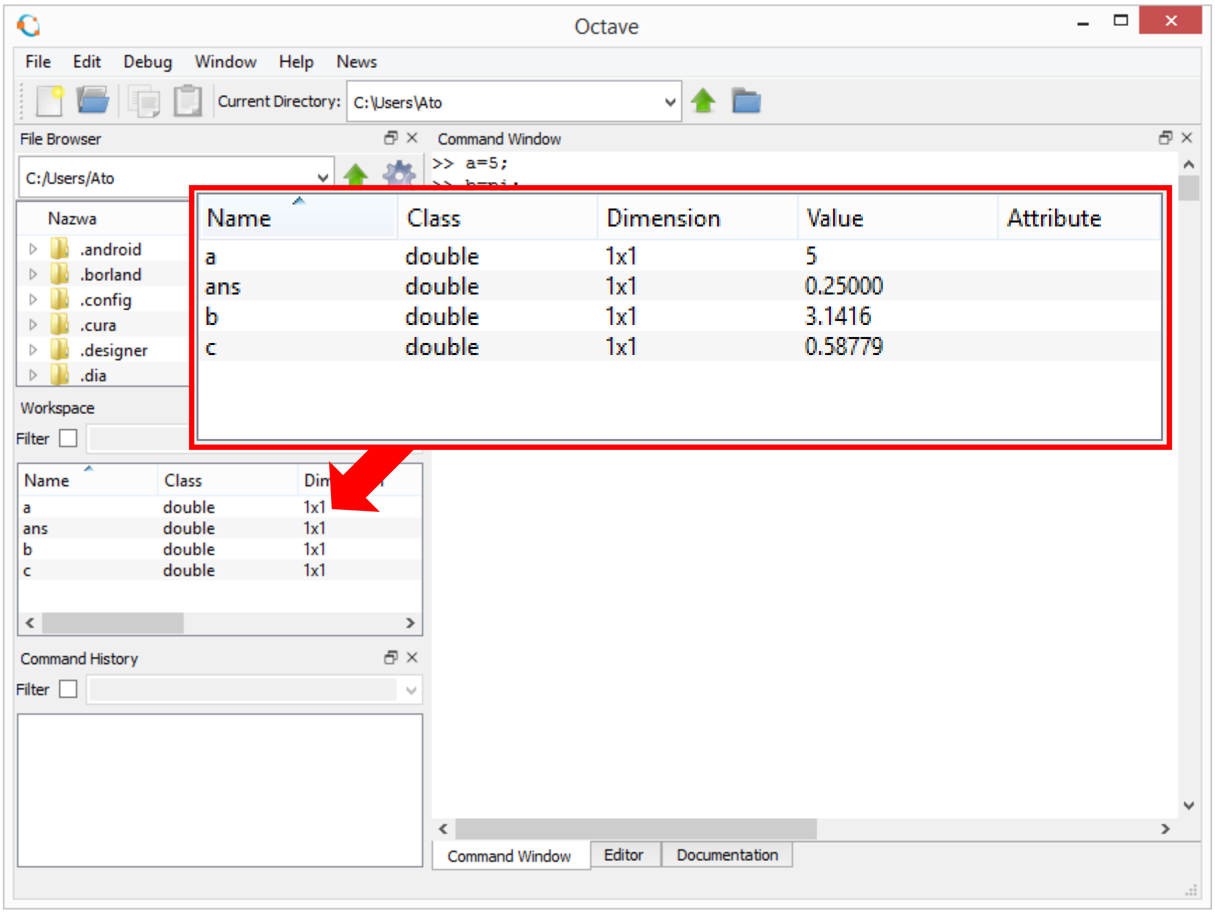
Εικ. 37 Το αποτέλεσμα της εντολής δεν εμφανίζεται.

Ομοίως, μπορούμε να δημιουργήσουμε πολλές μεταβλητές και να εκτελέσουμε ποικίλους υπολογισμούς (Εικόνα 38).



Εικ. 38 Διάφοροι υπολογισμοί

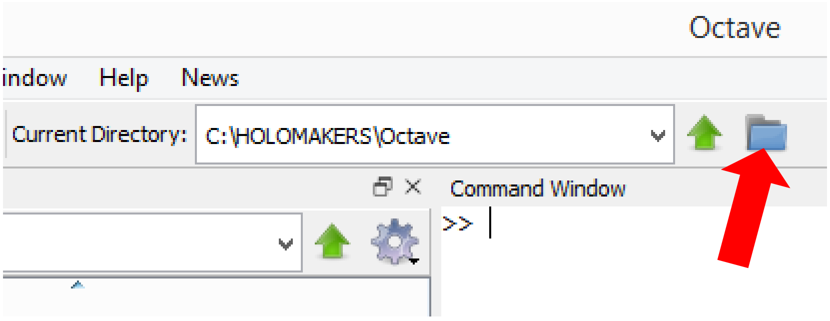
Αν κάποιος θέλει να δει όλες τις μεταβλητές που έχουν δηλωθεί μέχρι τώρα, αρκεί να κοιτάξει το παράθυρο "Workspace" (Εικόνα 39).



Εικ. 39 Το παράθυρο "Workspace"

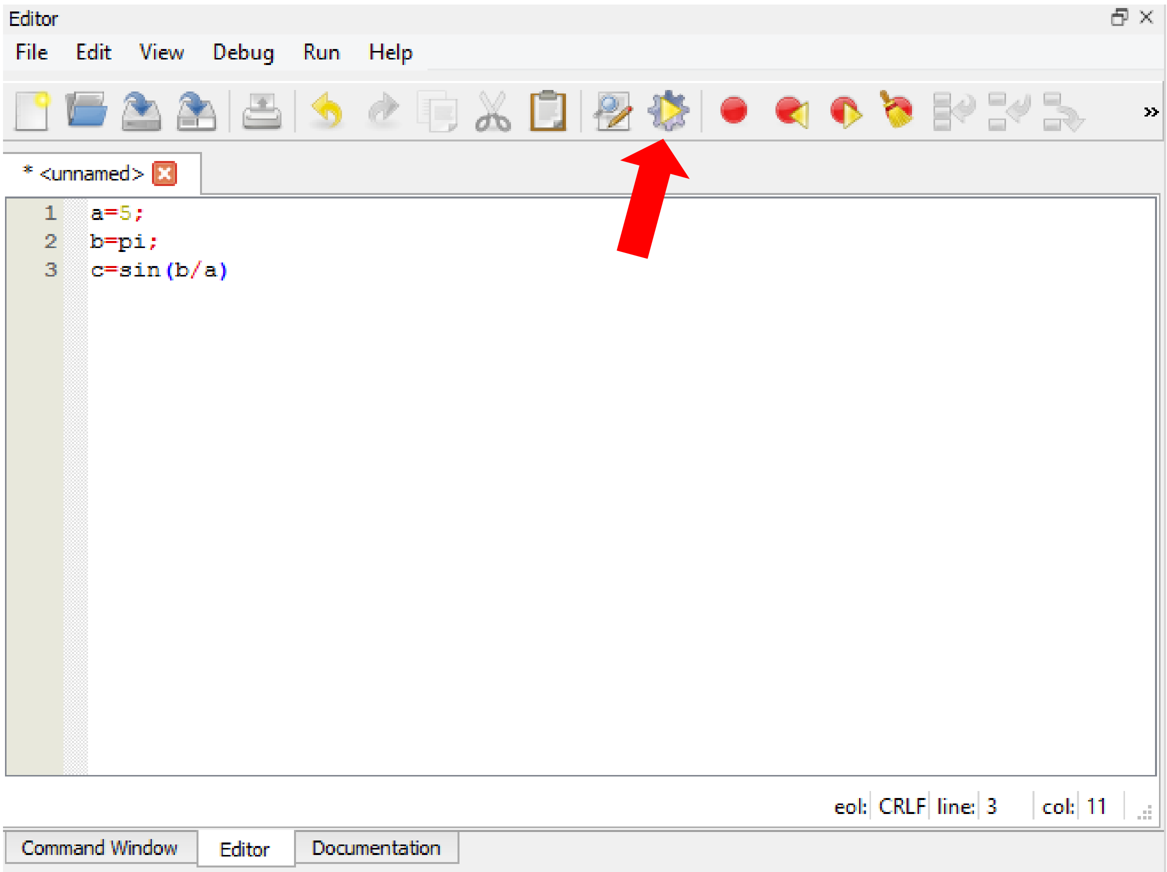
3.4. Editor

Όπως έχει ήδη αναφερθεί, ο δεύτερος τρόπος εισαγωγής εντολών είναι μέσω του Editor. Εδώ, θα πρέπει πρώτα να γραφούν όλες οι εντολές και μετά να «τρέξει» ο κώδικας. Πηγαίνετε στο "Editor" (Εικόνα 33) και εισάγετε τις εντολές (ή, όπως ονομάζεται, τον πηγαίο κώδικα). Πριν από αυτό όμως είναι καλό να ορίσετε τον φάκελο εντός του οποίου δουλεύετε. Για να το κάνετε αυτό, κάντε κλικ στο μπλε εικονίδιο, όπως φαίνεται στην εικόνα 40, και μετά επιλέξτε τον κατάλληλο φάκελο (για παράδειγμα τον φάκελο C: \ HOLOMAKERS \ Octave).



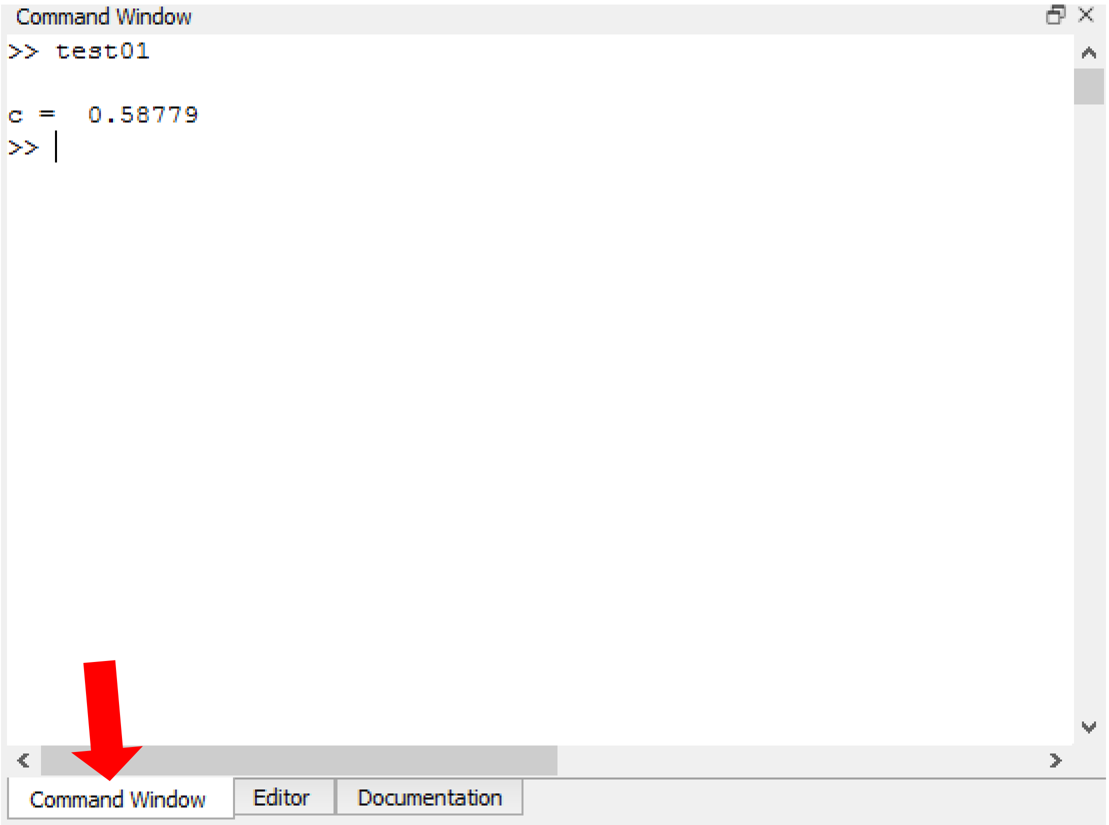
Εικ. 40 Αλλάζοντας τον φάκελο εργασίας

Ας γράψουμε τον δικό μας κώδικα. Η εικόνα 41 παρουσιάζει μερικές απλές εντολές. Κάθε εντολή είναι μια ξεχωριστή γραμμή (γεγονός που μας επιτρέπει να κρατήσουμε τον κώδικα ευανάγνωστο και να αποφύγουμε τα λάθη). Για να γίνουν οι υπολογισμοί θα πρέπει να τρέχουμε τον κώδικα χρησιμοποιώντας το εικονίδιο "Save File and Run", όπως υποδεικνύεται με το κόκκινο βέλος στην εικόνα 41. Την πρώτη φορά που θα εκτελέσετε την εντολή αυτή, το πρόγραμμα θα σας ζητήσει να σώσετε το αρχείο. Το αρχείο θα σωθεί στον φάκελο που δημιουργήσατε νωρίτερα. Δώστε στο αρχείο το όνομα "test01". Αυτό θα σωθεί με τη μορφή "test01.m". Έπειτα από αυτό το βήμα, ο κώδικας θα εκτελεστεί (με τη σειρά που οι εντολές είναι γραμμένες).



Εικ. 41 Editor

Για να δείτε τα αποτελέσματα των ενεργειών σας, επιστρέψτε στο "Command Window" Εικόνα 42). Μόνο η μεταβλητή "c" είναι ορατή λόγω του ότι η εξίσωση c = sin (b / a) μια και δεν είχε το ελληνικό ερωτηματικό στο τέλος. Οι δύο πρώτες εντολές που το είχαν δεν εμφανίζονται.



Εικ. 42 Τα αποτελέσματα των υπολογισμών

3.5. Προγραμματίζοντας στο Octave

3.5.1. Οι μεταβλητές

Οι μεταβλητές χρησιμοποιούνται για να αποθηκεύσουν δεδομένα που χρειαζόμαστε να βρίσκονται στη μνήμη του υπολογιστή (νούμερα, χαρακτήρες, κείμενα). Έτσι, μπορούμε να διαβάσουμε την τιμή μιας μεταβλητής οποιαδήποτε στιγμή. Στο Octave η δήλωση μιας μεταβλητής γίνεται πολύ εύκολα. Απλά γράφουμε το όνομα της μεταβλητής και ορίζουμε την τιμή της, με τον τρόπο που αναφέρθηκε σε προηγούμενη ενότητα. Στο Octave, μια μεταβλητή μπορεί να αποθηκεύσει πολλούς αριθμούς. Μια τέτοια μεταβλητή αποκαλείται πίνακας (array). Οι πίνακες μπορεί να είναι είτε μονοδιάστατοι είτε δισδιάστατοι (εικόνα 43).

Μονοδιάστατος πίνακας

a(1)	a(2)	a(3)	...	a(n)

δισδιάστατος πίνακας

a(1,1)	a(1,2)	a(1,3)	...	a(1,n)
a(2,1)	a(2,2)	a(2,3)	...	a(2,n)
a(3,1)	a(3,2)	a(3,3)	...	a(3,n)
...	...	...	...	...
a(m,1)	a(m,2)	a(m,3)	...	a(m,n)

Εικ. 43 διάταξη δεδομένων σε πίνακα

Όπως είναι φανερό, κάθε στοιχείο ενός μονοδιάστατου πίνακα ορίζεται από έναν δείκτη, ενώ στην περίπτωση του δισδιάστατου πίνακα ορίζεται από δύο δείκτες. Ας υποθέσουμε ότι θέλουμε να ορίσουμε μια μεταβλητή με το όνομα tab1, στην οποία θα αποθηκεύσουμε 5 νούμερα. Σε αυτή την περίπτωση θα γράψουμε:

```
tab1 = zeros(5,1)
```

Πρόκειται για έναν πίνακα που έχει 5 γραμμές και 1 στήλη. Κάθε ένα από τα 5 στοιχεία του πίνακα έχει μια τιμή ίση με 0. Ας ορίσουμε τώρα έναν δισδιάστατο πίνακα 3x3 ο οποίος να είναι επίσης γεμάτος με μηδενικά:

```
tab2 = zeros(3,3)
```

Εισάγεται το παραπάνω στο "Command Window" για να δείτε τους πίνακες που δημιουργήσατε.

Το ακόλουθο παράδειγμα είναι ενδιαφέρον ως προς τη δημιουργία ενός μονοδιάστατου πίνακα που αποτελείται από ισαπέχοντες αριθμούς ενός συγκεκριμένου εύρους. Ας δημιουργήσουμε έναν πίνακα που θα τον ονομάσουμε X, ο οποίος τα περιέχει 5 στοιχεία σε εύρος από 0 έως π. Για να το κάνουμε αυτό θα χρησιμοποιήσουμε τη λειτουργία linspace και θα εισάγουμε στοιχεία για την αρχή εύρους, το τέλος εύρους, και τον αριθμό των στοιχείων:

```
X=linspace(0,pi,5)
```

Γράψτε το παραπάνω στο "Command Window" για να δείτε τα αποτελέσματα.

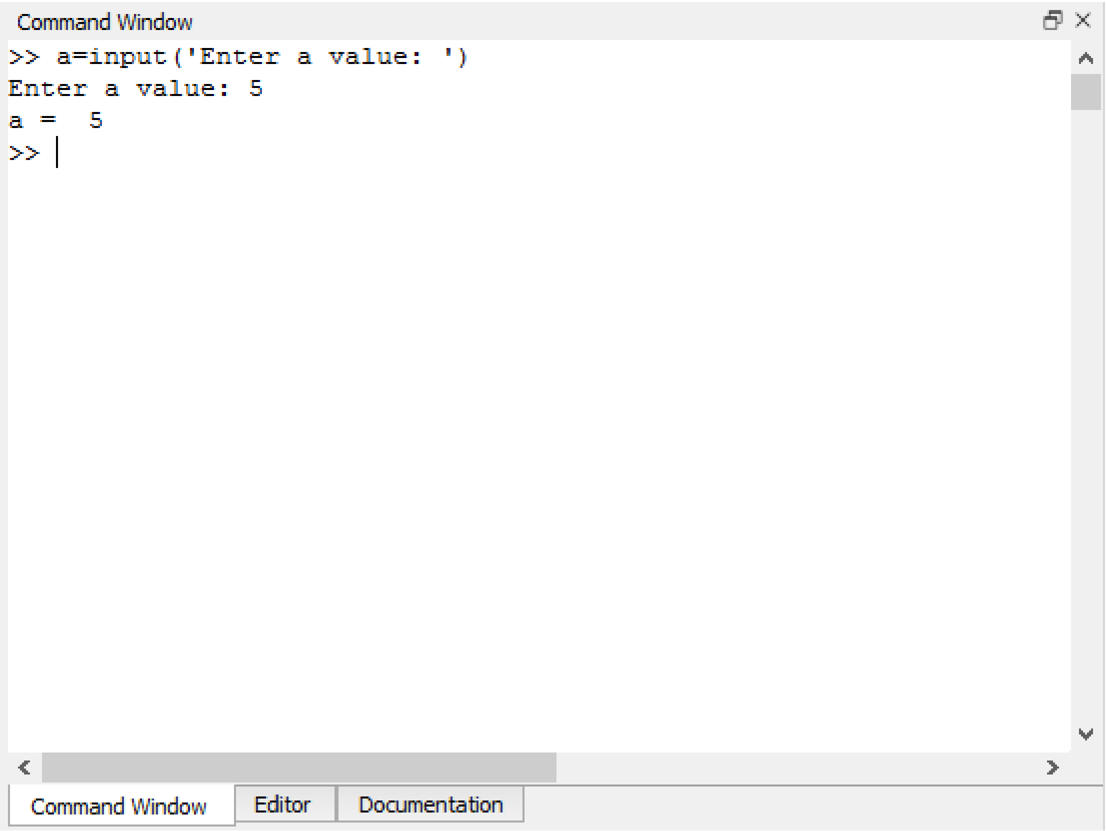
3.5.2. Βασικοί χειρισμοί στα εισαγόμενα και εξαγόμενα στοιχεία

Στο σημείο αυτό θα γίνει αναφορά στο χειρισμό εισαγωγής δεδομένων (input data) μέσω πληκτρολογίου. Ο χρήστης δίνει κάποιες τιμές μέσω πληκτρολογίου, τις οποίες θα ανακαλεί το πρόγραμμα στις κατάλληλες μεταβλητές. Τα εξαγόμενα στοιχεία (output) και χειρισμοί αφορούν την ένδειξη των υπολογισμών, π.χ. στην οθόνη.

Για να ορίσετε μια μεταβλητή με το όνομα "a", χρησιμοποιείστε την ακόλουθη εντολή: `a=input('User text')`. Πληκτρολογήστε το παρακάτω στο Command Window και επιβεβαιώστε πατώντας Enter.

```
a=input('Enter a value: ')
```

Στην επόμενη γραμμή θα δείτε το κείμενο: `Enter a value:` και το πρόγραμμα θα περιμένει να ορίσετε την τιμή πληκτρολογώντας την. Πληκτρολογήστε τον αριθμό 5 και επιβεβαιώστε με το Enter. Από εδώ και πέρα η μεταβλητή "a" θα έχει την τιμή 5. Το αποτέλεσμα θα μοιάζει με αυτό της εικόνας 44.



Εικ. 44 Εισάγοντας δεδομένα μέσω του πληκτρολογίου

Το απλούστερο αποτέλεσμα μια ενέργειας είναι να προβληθεί ένα κείμενο στην οθόνη με την εντολή `disp`. Γράψτε στο "Command Window" την επόμενη εντολή και πατήστε Enter:

```
disp('The variable "a" contains the value: '), disp(a)
```

Το ακόλουθο κείμενο θα προβληθεί: `The variable "a" contains the value:` και στην επόμενη γραμμή θα προβληθεί η τιμή της μεταβλητής "a". Έτσι, με την ενέργεια `disp` μπορείτε να προβάλετε στην οθόνη τόσο το κείμενο όσο και την τιμή της μεταβλητής. Αν θέλουμε να προβάλουμε μόνο το κείμενο, τότε μπορούμε επίσης να χρησιμοποιήσουμε την ενέργεια `puts`.

3.5.3. Χειριστές

Θα μιλήσουμε για χειριστές τριών τάξεων: τους αριθμητικούς, τους σχετικούς και τους λογικούς.

1. Οι αριθμητικοί χειριστές δεν είναι τίποτα άλλο παρά χειριστές πρόσθεσης (+), αφαίρεσης (-), πολλαπλασιασμού (*), διαίρεσης (/) και χειριστές που θέτουν τιμές (=). Έτσι, για παράδειγμα, η έκφραση a = c + 2 * b περιέχει τρεις χειριστές (=, +, *).

2. Οι σχετικοί χειριστές χρησιμοποιούνται για να συγκρίνουν δύο αξίες (π.χ., a> b). Αυτοί οι χειριστές είναι: χειριστές ισότητας (==), μικρότεροι από (<), μεγαλύτεροι από (>), λιγότεροι ή ίσοι του (<=), μεγαλύτεροι οι ίσοι του (> =), διάφοροι από (! =). Το αποτέλεσμα της σύγκρισης δύο τιμών είναι η ονομαζόμενη λογική τιμή, η οποία είναι είτε αληθής είτε ψευδής. Για παράδειγμα, αν γράψουμε την έκφραση 2 == 5, θα είναι ψευδής, ενώ η 2 <5 θα είναι αληθής.

3. Οι λογικοί χειριστές χρησιμοποιούνται για να πραγματοποιούν βασικούς λογικούς χειρισμούς όπως ένα λογικό άθροισμα (||), μια λογική σύνδεση (&&) καθώς και αρνητικούς χειρισμούς (!). Το αποτέλεσμα μια λογικής έκφρασης είναι είτε σωστό είτε λάθος. Για παράδειγμα, η έκφραση (5> 2) && (2 <3) είναι σωστή επειδή τόσο το 5> 2 όσο και το 2 <3 είναι και τα δύο αληθινά. Η συγκεκριμένη έκφραση χρησιμοποιεί και σχετικούς χειριστές (<,>) και λογικούς (&&).

Μια βασική ιδέα που αξίζει να τονισθεί είναι η λεγόμενη προτεραιότητα χειριστών. Σκεφτείτε για παράδειγμα την εξής αλληλουχία: 5 + 2 * 3 = ?. Το αποτέλεσμα είναι 11 γιατί γνωρίζουμε απ’ τα μαθηματικά ότι ο πολλαπλασιασμός προηγείται της πρόσθεσης. Το ίδιο κάνουν και οι χειριστές προτεραιότητας. Μας λένε ποιοι χειρισμοί προηγούνται και ποιοι έπονται. Ο πίνακας 1 δείχνει τους χειριστές με τη σειρά προτεραιότητας. Η λογική άρνηση έχει την υψηλότερη προτεραιότητα. Έτσι, όπου συναντάμε αυτό τον χειριστή τον εκτελούμαι πρώτο. Οι σχετικοί χειριστές έχουν τη μικρότερη προτεραιότητα, και συνεπώς εκτελούνται στο τέλος.

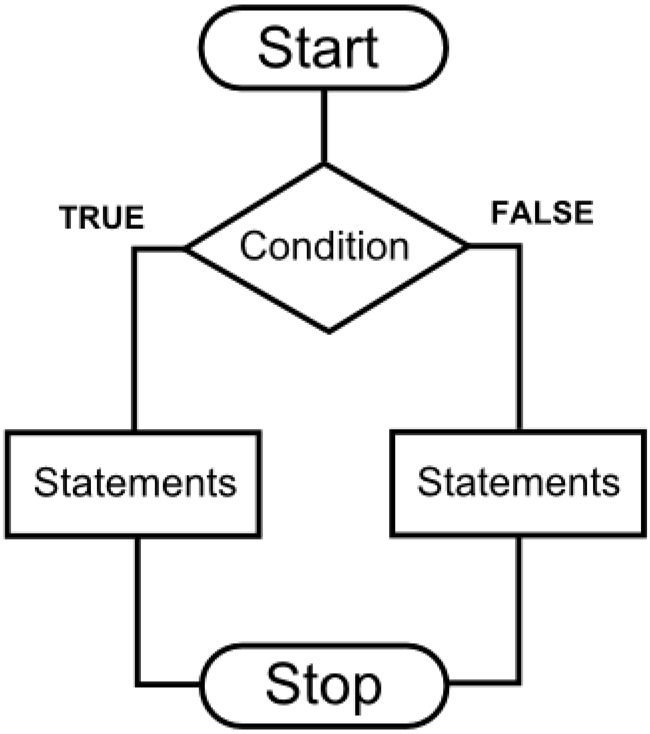
Priority	Operator
1	!
2	&&, *, /
3	||, +, -
4	==, <, >, <=, >=, !=

Table 1 Προτεραιότητα χειριστών

Κάτι άλλο που μας είναι γνωστό από τα μαθηματικά είναι ότι αν θέλουμε να αλλάξουμε τη σειρά βάσει της οποίας γίνονται οι υπολογισμοί, πρέπει να χρησιμοποιήσουμε παρενθέσεις. Έτσι, η εξίσωση 5 + 2 * 3 δίνει τιμή 11, ενώ η εξίσωση (5 + 2) * 3 θα δώσει την τιμή 21. Μερικές φορές η εισαγωγή παρενθέσεων είναι αναγκαία ώστε οι εξισώσεις να έχουν νόημα. Ένα τέτοιο παράδειγμα έχει ήδη δοθεί παραπάνω: (5> 2) && (2 <3).

3.5.4. Υποθετική/ Υπό συνθήκες δήλωση

Όλοι μας παίρνουμε διαφορετικές αποφάσεις σε καθημερινή βάση. Για παράδειγμα, αν έχει ήλιο, αποφασίζουμε ότι θα πάμε για ποδήλατο. Αυτό είναι ένα παράδειγμα μιας απλής δήλωσης υπό συγκεκριμένες συνθήκες. Αρχικά λοιπόν ελέγχουμε ορισμένες συνθήκες (π.χ. τον καιρό), και μετά, ανάλογα με το αποτέλεσμα της συνθήκης, δρούμε (π.χ. πάμε για ποδήλατο). Η συνθήκη αναπαρίσταται από μια λογική έκφραση που είναι είτε αληθής είτε ψευδής. Μια λογική έκφραση μπορεί για παράδειγμα να είναι «Τώρα ο καιρός είναι αίθριος». Αυτή η λογική έκφραση είναι είτε αληθής είτε ψευδής, ανάλογα με τις καιρικές συνθήκες.
Στον προγραμματισμό, οι υπό συνθήκες δηλώσεις είναι τόσο απαραίτητες όσο και τα αντίστοιχα παραδείγματα στην πραγματική ζωή. Η εικόνα 45 παρουσιάζει τη δομή μιας απλής οδηγίας, υπό συνθήκες. Ανάλογα με το αν η συνθήκη είναι αληθής ή όχι, εκτελούνται άλλες οδηγίες (δηλώσεις).

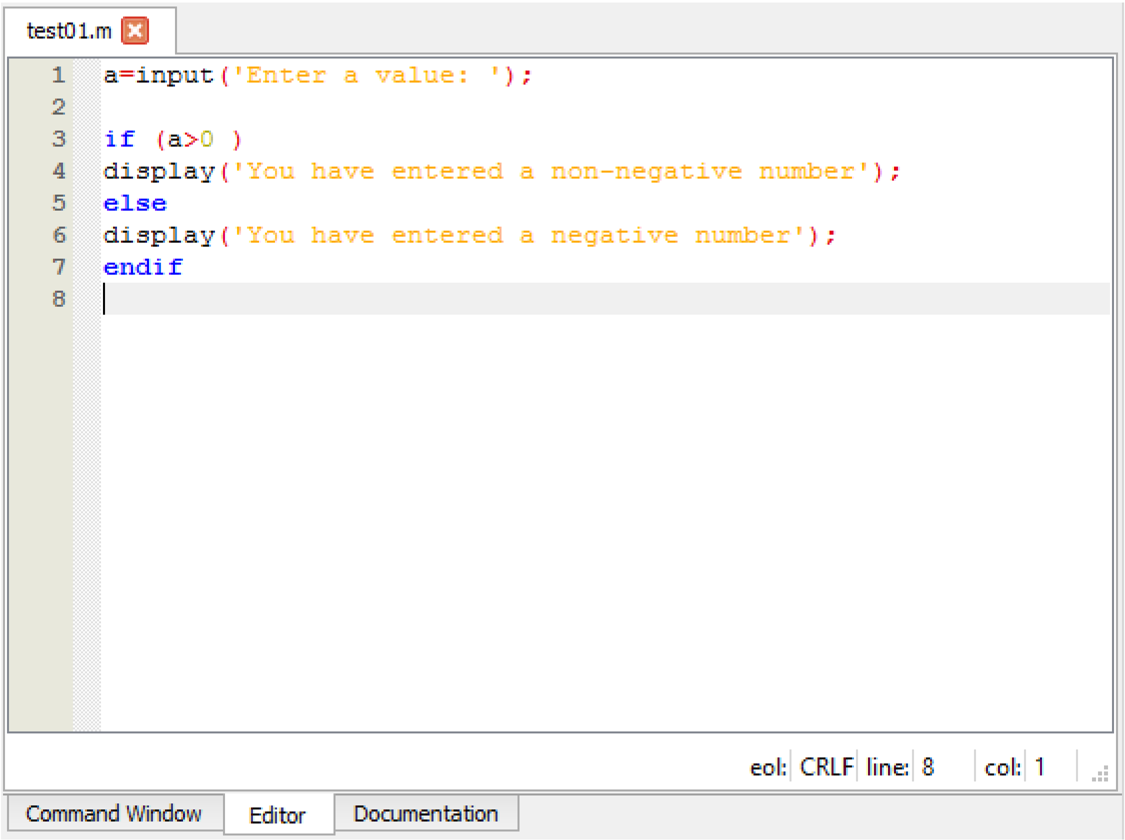


Εικ. 45 Αλγόριθμος μιας απλής, υπό συνθήκες, δήλωσης

Στο Octave, απλές, υπό συνθήκες, δηλώσεις, μοιάζουν με τις επόμενες:

```
if (condition)
Άλλες δηλώσεις που θα εκτελεστούν αν η συνθήκη είναι αληθής
else
Άλλες δηλώσεις που θα εκτελεστούν αν η συνθήκη είναι ψευδής
endif
```

Στη συνέχεια θα γράψουμε έναν σύντομο κώδικα που θα ελέγχει αν ο αριθμός που εισάγεται είναι αρνητικός ή όχι, ως ένα παράδειγμα της χρήσης της υπό συνθήκες δήλωσης. Στο παράθυρο Editor γράψτε τον κώδικα που φαίνεται στην εικόνα 46. Μετά κάντε κλικ στο εικονίδιο "save file and run" και πηγαίνετε στο Command Window. Αφού εισάγεται το νούμερο και πατήσετε το Enter, η αντίστοιχη πρόταση θα εμφανιστεί.



Εικ. 46 Παράδειγμα χρήσης της υπό συνθήκες δήλωσης

Συχνά ωστόσο, αντιμετωπίζουμε περιπτώσεις όπου θέλουμε να τσεκάρουμε παραπάνω από μια συνθήκες. Έτσι στο Octave πρέπει να γράψουμε:

`if (condition1)`
Άλλες δηλώσεις που θα εκτελεστούν αν η συνθήκη `condition1` είναι αληθής
`elseif (condition2)`
Άλλες δηλώσεις που θα εκτελεστούν αν η συνθήκη `condition1` είναι ψευδής και η `condition2` είναι αληθής
`else`
Άλλες δηλώσεις που θα εκτελεστούν αν και η `condition1` και η `condition2` είναι ψευδής
`endif`

Για να το κατανοήσουμε καλύτερα, ας τροποποιήσουμε τον τελευταίο κώδικα έτσι ώστε το πρόγραμμα να δείχνει αν ο χρήστης έχει εισάγει το νούμερο μηδέν (εικόνα 47).

```
a=input('Enter a value: ');

if (a>0 )
display('You have entered a positive number');
elseif (a<0)
display('You have entered a negative number');
else
display('You have entered 0');
endif
```

Εικ. 47 Παράδειγμα της υπό συνθήκες δήλωσης

3.5.5. Η επανάληψη (loop) "For"

Συχνά υπάρχει η ανάγκη να εκτελέσουμε κάποιες εντολές, πολλαπλές φορές. Για να αποφύγουμε την εισαγωγή της ίδιας εντολής πολλές φορές, χρησιμοποιούμε την επανάληψη (loop). Εδώ θα περιγράψουμε μόνο την επανάληψη "for" η οποία επιτρέπει την εκτέλεση μιας σειράς εντολών, για ένα συγκεκριμένο αριθμό φορών. Η δομή της εντολής φαίνεται στην εικόνα 48. Είναι πολύ απλή και μας λέει ότι η συγκεκριμένη δήλωση θα επαναληφθεί – στη συγκεκριμένη περίπτωση - 10 φορές. Πώς ξέρουμε όμως ότι θα γίνει μόνο 10 φορές; Αυτό ορίζεται στην αρχή της επανάληψης. Μια μεταβλητή, που ονομάζεται "i" εφαρμόζεται σε κάθε επόμενη εκτέλεση, και αυξάνει την τιμή ανά 1 μονάδα, από το 1 έως το 10. Όταν η μεταβλητή "i" φτάσει στο τέλος (τιμή 10) η εντολή της επανάληψης σταματά.

```
for i=1:10

statements...

end
```

Εικ. 48 Η επανάληψη "For"

Ένα πολύ απλό παράδειγμα θα μπορούσε να είναι η εμφάνιση ενός αριθμού χαρακτήρων στην οθόνη. Φανταστείτε ότι θέλουμε να μπορούμε να εμφανίσουμε μια γραμμή που αποτελείται από τον χαρακτήρα "-". Μπορούμε να γράψουμε έναν απλό κώδικα, όπως φαίνεται στην εικόνα 49. Δημιουργείστε έναν νέο κώδικα κάνοντας κλικ στο εικονίδιο . Κατά τη διάρκεια της πρώτης εκτέλεσης, μπορείτε να σώσετε το αρχείο με το όνομα test02. Όπως μπορείτε να δείτε η γραμμή μας αποτελείται από 15 χαρακτήρες (η επανάληψη "for" θα εκτελεστεί 15 φορές).

```
for i=1:15

  puts('-');

  end
```

Εικ. 49 Δημιουργώντας μια γραμμή 15 χαρακτήρων

Το αποτέλεσμα του κώδικα μπορεί να φανεί στο Command Window (Εικόνα 50).

```
Command Window
>> test02

---------------->>
```

Εικ. 50 Μια γραμμή 15 χαρακτήρων "-"

Μπορούμε να τροποποιήσουμε τον κώδικα ώστε να επιτρέπει στον χρήστη να αποφασίσει το μήκος της γραμμής (Εικόνα 51). Η επανάληψη θα εκτελεστεί για n φορές, όπου n είναι ένας αριθμός που παρέχεται από τον χρήστη.

```
n=input ('enter the length of the line: ');

for i=1:n

puts('-');

end
```

Εικ. 51 Δημιουργώντας μια γραμμή χαρακτήρων ενός τυχαίου αριθμού

Η επανάληψη for χρησιμοποιείται συχνά για ποικίλων ειδών υπολογισμούς. Μπορείτε να δημιουργήσετε έναν κώδικα με τον οποίο θα υπολογίζεται τη μέση τιμή 5 αριθμών; Αν πιστεύετε πώς όχι, δείτε την εικόνα 52.

```
for i=1:5

a(i) = input('Enter a value: ');

end

sum = 0;

for i = 1:5

sum = sum + a(i);

end

average = sum/5
```

Εικ. 52 Υπολογισμός του αριθμητικού μέσου

Για να δούμε τι γράψαμε: Πρώτα εισαγάγαμε 5 αριθμούς. Οι αριθμοί αυτοί γράφονται σε έναν πίνακα (array) που ονομάζεται "a". Στη συνέχεια, καλούμε τη μεταβλητή με το όνομα sum, της οποίας την τιμή θέτουμε στο 0. Ύστερα, η επανάληψη εκτελεί το άθροισμα των 5 αριθμών που εισαγάγαμε. Η τελευταία γραμμή αφορά τον υπολογισμό του αριθμητικού μέσου. Αν δεν εισάγουμε το ελληνικό ερωτηματικό στο τέλος, το αποτέλεσμα της πράξης θα φανεί στην οθόνη.

3.5.6. Γραφήματα

Το Octave επιτρέπει τη δημιουργία του γραφήματος μιας συνάρτησης με έναν πολύ απλό τρόπο. Για να δημιουργήσουμε το γράφημα του $y = f(x)$ πρέπει πρώτα να δημιουργήσουμε έναν πίνακα με ανεξάρτητες μεταβλητές για το x και για το $f(x)$. Έστω ότι θέλουμε να σχεδιάσουμε τη συνάρτηση $y = \sin(x)$ σε ένα εύρος $[0,2\pi]$. Το Octave θα μας επιτρέψει να δηλώσουμε έναν πίνακα με n ισ-απέχοντα στοιχεία στο διάστημα $[a,b]$. Αυτό γίνεται με τη χρήση της συνάρτησης `linspace(a, b, n`. Ας υποθέσουμε ότι θέλουμε να χωρίσουμε το εύρος $[0, 4\pi]$ σε 100 ισαπέχουσες τιμές. Σε αυτή την περίπτωση πρέπει να γράψουμε:

```
x = linspace(0, 4*pi,100);
```

Έτσι, δηλώνουμε τον πίνακα των 100 στοιχείων. Το να δημιουργήσουμε τον πίνακα των τιμών μιας εξίσωσης είναι ακόμα πιο εύκολο, καθώς αρκεί να γράψουμε:

```
y=sin(x);
```

Λόγω του ότι το x είναι ένας πίνακας 100 στοιχείων, το y θα μετατραπεί αυτομάτως σε έναν πίνακα με τον ίδιο αριθμό στοιχείων. Τώρα μπορούμε να απεικονίσουμε το γράφημα. Για να το κάνουμε αυτό απλά γράφουμε:

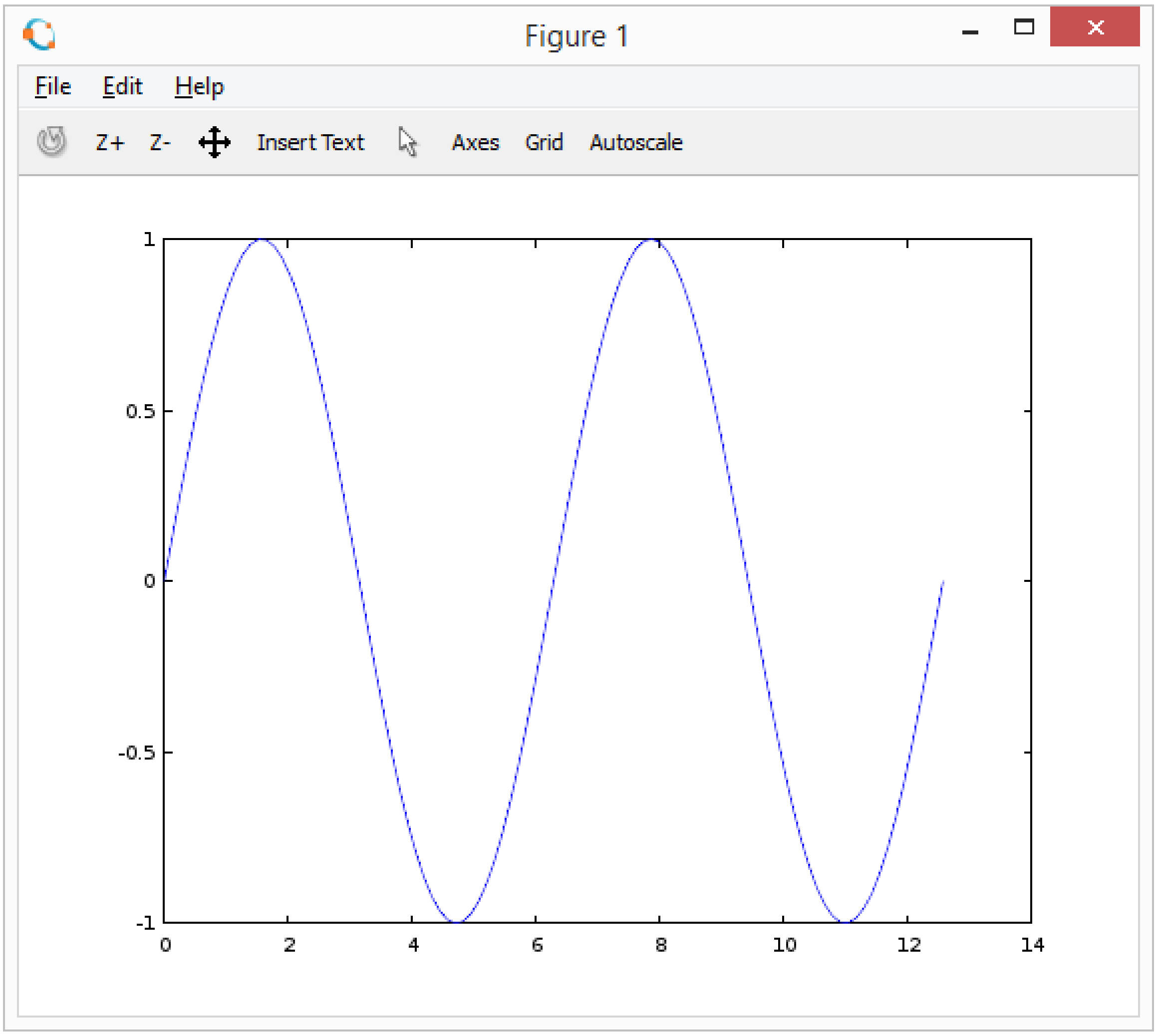
```
plot(x, y)
```

Ολόκληρος ο κώδικας φαίνεται στην εικόνα 53.

```
x = linspace(0, 4*pi,100);
y = sin(x);
plot(x, y)
```

Εικ. 53 Προετοιμασία δεδομένων και γραφική απεικόνιση

Μετά την εκκίνηση του κώδικα ένα νέο παράθυρο ανοίγει που περιέχει το σχετικό γράφημα (εικόνα 54).

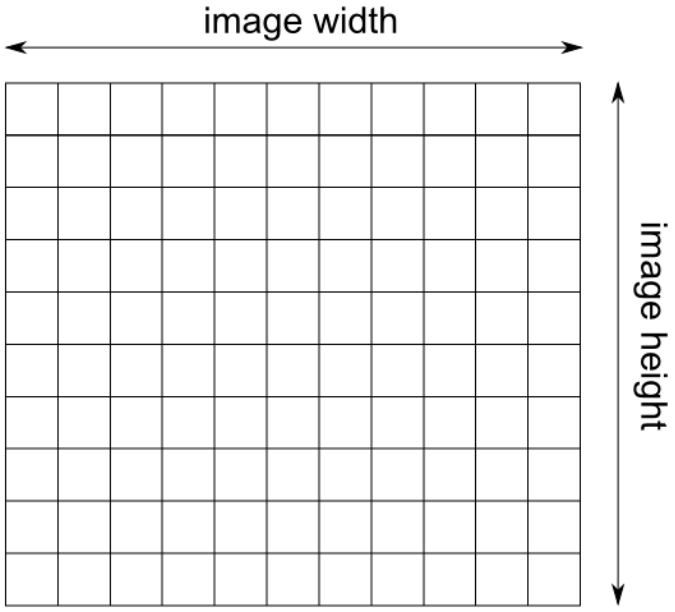


Εικ. 54 Γράφημα

3.6. Επεξεργασία εικόνας (Image processing)

3.6.1. Δημιουργώντας μια εικόνα

Στο Octave οι εικόνες (γκρι κλίμακας) αναπαρίστανται ως δισδιάστατοι πίνακες, όπου κάθε στοιχείο αντιστοιχεί σε ένα pixel (εικόνα 55).



Εικ. 55

Επομένως, η δημιουργία νέων εικόνων αφορά απλά τη δήλωση ενός δισδιάστατου πίνακα. Αν θέλουμε να δημιουργήσουμε μια εικόνα μεγέθους 200x200 pixels, απλά γράφουμε:

```
image1 = zeros(200,200);
```

Με αυτό τον τρόπο η μεταβλητή `image1` ορίζεται ως ένας πίνακας 200x200 γεμάτος με μηδενικά (κάθε στοιχείο του πίνακα είναι μηδέν).

3.6.2. Διαβάζοντας/ γράφοντας μια εικόνα και εμφανίζοντάς τη στην οθόνη

Συχνά υπάρχει η ανάγκη να φορτώσουμε ένα ήδη υπάρχον αρχείο εικόνας. Για να το κάνουμε αυτό, αντιγράφουμε πρώτα την εικόνα στον φάκελο που δουλεύουμε ("current directory"). Μετά απλά γράφουμε την εντολή:

```
image1 = imread('filename');
```

Το αρχείο της εικόνας είναι τώρα αποθηκευμένο στη μεταβλητή image1.
Για να εμφανίσουμε την εικόνα χρησιμοποιούμε απλά την εντολή:

```
imshow(image1);
```

Αν θέλουμε να προβάλουμε αρκετές εικόνες, θα πρέπει κάθε εικόνα να έχει οριστεί με έναν διαφορετικό αριθμό, χρησιμοποιώντας την εντολή `figure(numer)`. Αξίζει επίσης να ονομάσουμε και το κάθε παράθυρο που θα εμφανιστεί η εκάστοτε εικόνα, χρησιμοποιώντας την εντολή `title('Image Title')`. Παρακάτω είναι ένα παράδειγμα προβολής δύο διαφορετικών εικόνων, σε δύο διαφορετικά παράθυρα:

```
figure(1);imshow(image1);title('Image no 1');
figure(2);imshow(image2);title('Image no 2');
```

Η εικόνα σώζεται σε ένα αρχείο με τη χρήση της εντολής:

```
imwrite(image1,'name.extension');
```

Η εικόνα μπορεί να σωθεί σε διάφορους αρχειακούς τύπους/επεκτάσεις, π.χ. bmp, png, jpg, tiff, gif.Δοκιμάστε να ανεβάστε στον φάκελο εργασίας ένα αρχείο εικόνας με την επέκταση bmp, και έπειτα γράψτε έναν κώδικα που θα προβάλλει την εικόνα και θα τη σώσει με διαφορετικές επεκτάσεις

```
image1=imread('test.bmp');
imshow(image1);
imwrite(image1,'test.gif');
imwrite(image1,'test.tiff');
imwrite(image1,'test.png');
imwrite(image1,'test.jpg');
```

Εικ. 56 Μετατρέποντας την εικόνα σε διαφορετικούς τύπους/προεκτάσεις

3.6.3. Μετατροπή χρώματος

Το Octave έχει πολλές διαφορετικές λειτουργίες για την μετατροπή χρώματος. Μεταξύ αυτών υπάρχει μια ενέργεια που μετατρέπει την έγχρωμη εικόνα στην κλίμακα του γκρι, και μία άλλη που μετατρέπει οποιουδήποτε χρώματος εικόνα, σε δυαδική, που συντίθεται από δύο μόνο χρώματα (μαύρο και άσπρο). Αυτές οι ενέργειες ωστόσο δεν είναι στις εναρκτήριες διεργασίες του Octave. Για να είναι δυνατή η χρήση τους θα πρέπει να φορτωθεί το πακέτο αρχείων, με το όνομα `image`. Έτσι, στην αρχή του κώδικα γράφουμε:

```
pkg install image
pkg load image
```

Τώρα μπορούμε να χρησιμοποιήσουμε όλες τις διαθέσιμες λειτουργίες. Για να μετατρέψουμε μια έγχρωμη εικόνα σε γκρι, χρησιμοποιούμε την ενέργεια `rgb2gray`. Για παράδειγμα, αν η μεταβλητή `colorImage` αποθηκεύει μια έγχρωμη εικόνα, μπορούμε να τη μετατρέψουμε σε γκρι, και να τη σώσουμε στη μεταβλητή `grayImage`:

```
grayImage= rgb2gray(colorImage);
```

Παρόμοια, η εικόνα μπορεί να μετατραπεί σε δυαδική:

```
bwImage= im2bw(colorImage,0.5);
```

3.6.4. Περιστροφή

Μπορείτε να περιστρέψετε οποιαδήποτε εικόνα, με σημείο περιστροφής το κέντρο, χρησιμοποιώντας το `imrotate`. Αυτό που χρειάζεται είναι να υποδείξουμε την εικόνα που θα περιστραφεί και τη γωνία περιστροφής. Για παράδειγμα, αν θέλουμε να περιστρέψουμε την `Image1` σε γωνία 45 °, στη μεταβλητή `Image2` θα γράψουμε:

```
Image2 = imrotate(Image1,45);
```

3.6.5. Πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση.

Όπως θυμόμαστε, η εικόνα στο Octave αναπαρίσταται ως ένας δισδιάστατος πίνακας με διαστάσεις αντίστοιχες του πλάτους και του ύψους της εικόνας, σε pixels. Έχοντας δύο εικόνες με την ίδια διάσταση, μπορούμε εύκολα να εκτελέσουμε ενέργειες πρόσθεσης ή πολλαπλασιασμού εικόνων. Για παράδειγμα, για την πρόσθεση δύο εικόνων θα χρειαστεί απλά η πρόσθεση των αντίστοιχων pixel (Εικόνα 57).

$$
\underset{\textbf{Image A}}{\begin{matrix} 8 & 3 & 0 & 8 & 14 \\ 18 & 9 & 10 & 13 & 0 \\ 17 & 9 & 10 & 7 & 6 \\ 9 & 19 & 17 & 4 & 19 \\ 7 & 1 & 16 & 19 & 7 \end{matrix}}
+
\underset{\textbf{Image B}}{\begin{matrix} 20 & 12 & 2 & 19 & 15 \\ 15 & 11 & 5 & 17 & 15 \\ 16 & 6 & 5 & 9 & 11 \\ 8 & 15 & 13 & 19 & 17 \\ 8 & 0 & 9 & 5 & 3 \end{matrix}}
=
\underset{\textbf{Image C}}{\begin{matrix} 28 & 15 & 2 & 27 & 29 \\ 33 & 20 & 15 & 30 & 15 \\ 33 & 15 & 15 & 16 & 17 \\ 17 & 34 & 30 & 23 & 36 \\ 15 & 1 & 25 & 24 & 10 \end{matrix}}
$$

Εικ. 57 Προσθέτοντας δύο εικόνες

Με έναν παρόμοιο τρόπο γίνεται τόσο η αφαίρεση όσο και ο πολλαπλασιασμός και η διαίρεση. Ο πίνακας 2 παρουσιάζει τα ονόματα των συγκεκριμένων ενεργειών.

Operation	Octave
Add	`imadd`
Substract	`imsubstract`
Multiply	`immultiply`
Division	`imdivide`

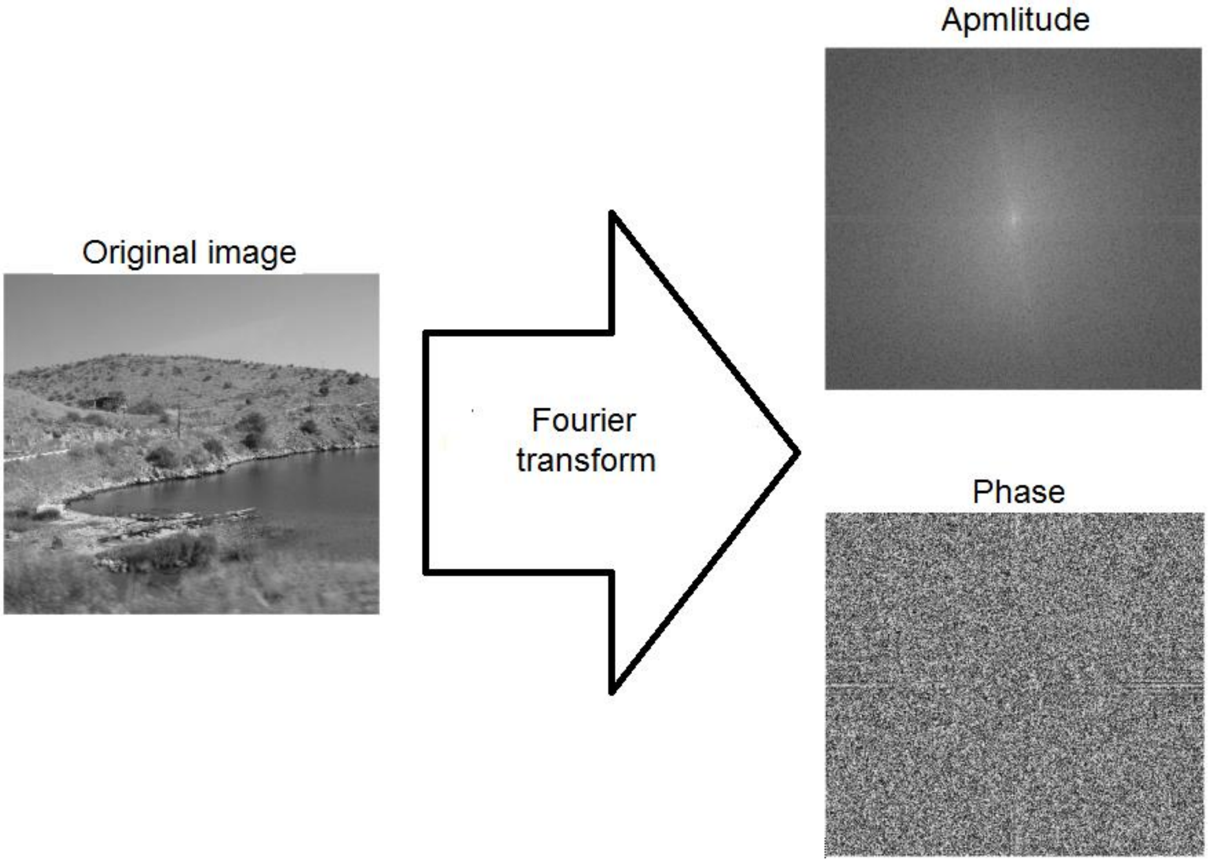
Table 2 Χειρισμοί ενεργειών στο Octave

Έστω ότι οι μεταβλητές `imgA` και `imgB` αναπαριστούν δύο εικόνες ίδιου μεγέθους. Μπορούμε να δημιουργήσουμε την imgC, η οποία θα είναι το άθροισμα ή το γινόμενό τους, με τους εξής τρόπους:

```
imgC = imadd(imgA,imgB);
imgC = immultiply(imgA,imgB);
```

3.6.6. Μετατροπή Fourier

Η μετατροπή Fourier είναι ένας τρόπος που χρησιμοποιείται ευρέως, σε ποικίλα πεδία, όπως η οπτική και η μουσική, και ανήκει στη διαδικασία επεξεργασίας εικόνας. Η συγκεκριμένη μετατροπή μοιάζει με τη συλλογή τυχαίων pixels που όμως εμπεριέχει τις ίδιες πληροφορίες με την αρχική εικόνα (Εικόνα 58).

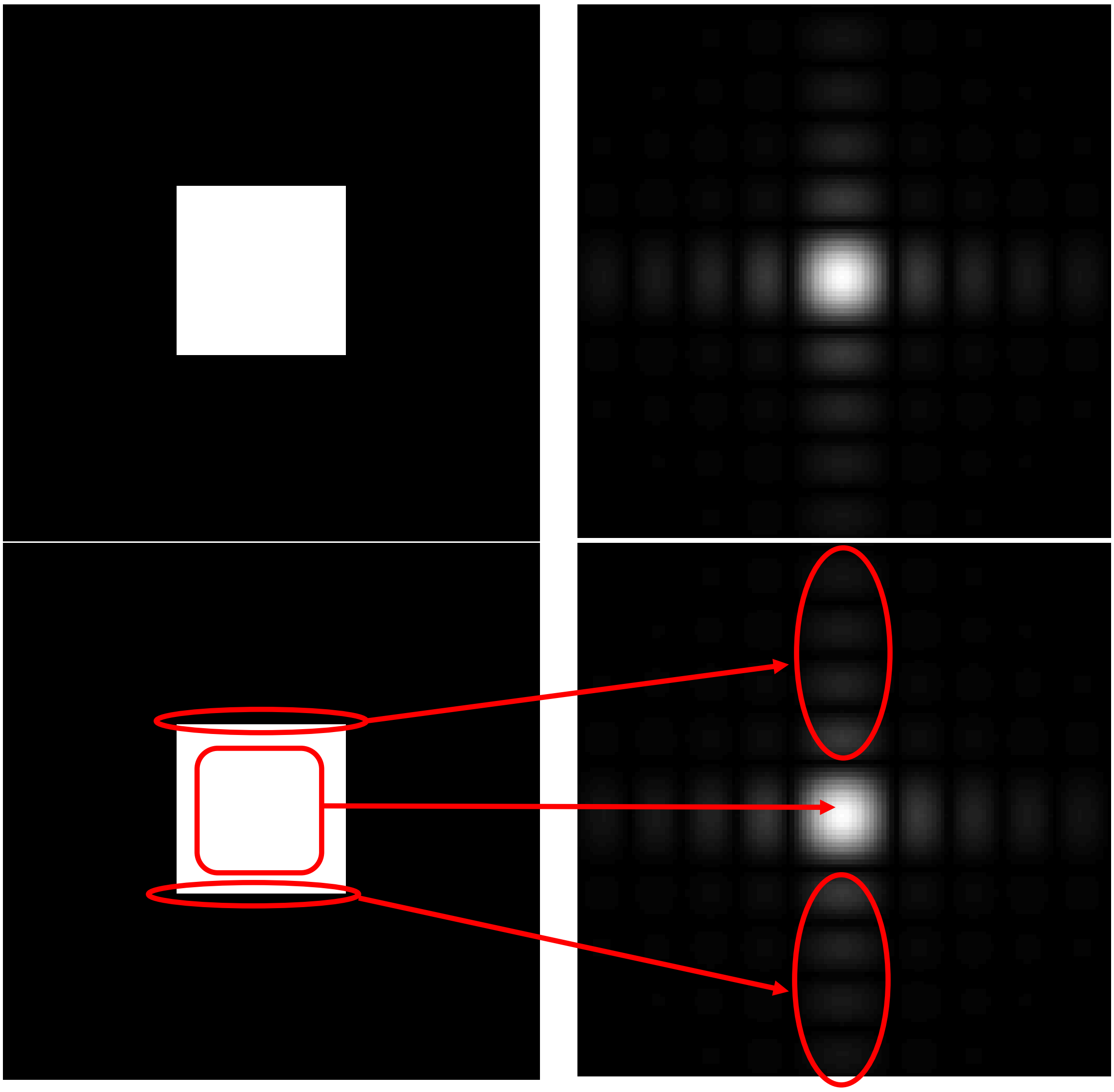


Εικ. 58 Η μετατροπή Fourier μιας εικόνας

Κάθε σημείο της μετατροπής Fourier δεν έχει μόνο πλάτος αλλά και φάση. Περισσότερες λεπτομέρειες επί του συγκεκριμένου θέματος όμως δεν συνάδουν με τον σκοπό του συγκεκριμένου οδηγού. Ας θυμηθούμε απλά ότι η φάση είναι απαραίτητη για την αναπαραγωγή της αρχικής εικόνας (μέσω της αντίστροφης μετατροπής Fourier), ενώ το πλάτος διαδίδεται με τους εξής τρόπους: τα σημεία στο κέντρο της μετατροπής αντιστοιχούν σε ομαλές περιοχές της εικόνας, ενώ τα σημεία που απομακρύνονται από το κέντρο, αφορούν τις ακμές της εικόνας (οι ακμές είναι για παράδειγμα το όριο μεταξύ ουρανού και βουνών, υγρού στοιχείου και ξηράς κλπ.). Η εικόνα 59, για παράδειγμα, παρουσιάζει ένα λευκό τετράγωνο και την αντίστοιχη μετατροπή Fourier. Το μεσαίο σημείο της μετατροπής Fourier αντιστοιχεί στο κεντρικό τμήμα του τετραγώνου, ενώ κάθε μία από τις ανακλώμενες περιοχές αντιστοιχεί σε μια εκ των ακμών του τετραγώνου (Εικόνα 59β).

α)

β)

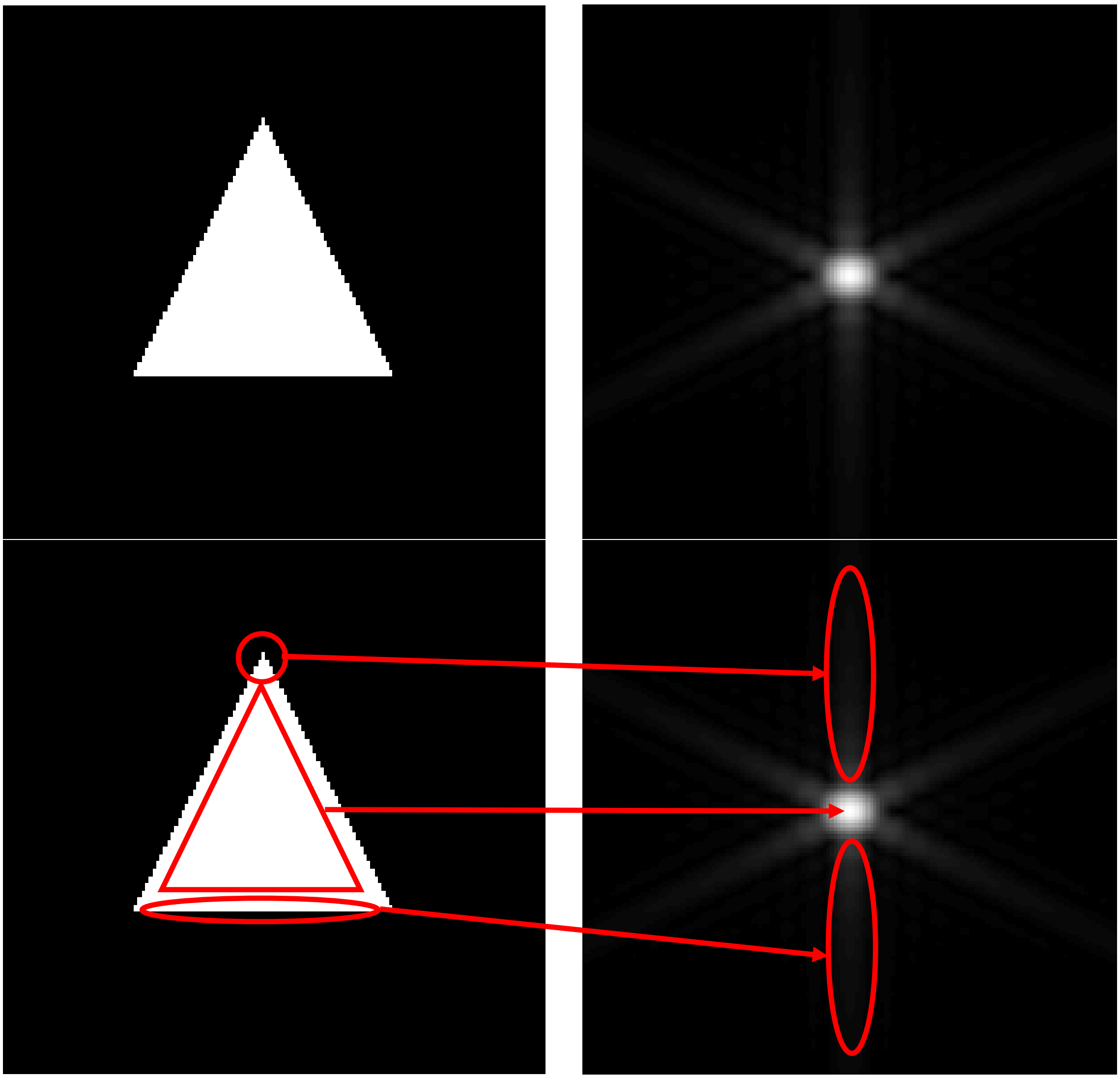


Εικ. 59 Η μετατροπή Fourier ενός τετραγώνου

Στην περίπτωση απλών γεωμετρικών φιγούρων το φαινόμενο είναι εύκολα ορατό. Η εικόνα 60 αναπαριστά ένα τρίγωνο και τη μετατροπή του. Το μεσαίο σημείο της μετατροπής αντιστοιχεί στο κέντρο του τριγώνου και τα τρία ανακλόμενα μονοπάτια στις πλευρές του. Στη συγκεκριμένη περίπτωση έχουμε και άλλα 3 ανακλόμενα μονοπάτια ορατά. Τα συγκεκριμένα απεικονίζουν τις κορυφές του τριγώνου.

a)

b)



Εικ. 60 Η μετατροπή Fourier ενός τριγώνου

Στο Octave, η ενέργεια της μετατροπής Fourier ονομάζεται `fft2`. Αυτή η ενέργεια λειτουργεί με τέτοιο τρόπο που η μετατροπή Fourier έχει μια μετατόπιση από το κέντρο. Γι' αυτό είναι απαραίτητη και η χρήση της ενέργειας `fftshift`. Αν λοιπόν θέλουμε να δημιουργήσουμε τη μετατροπή Fourier της `img1` θα γράψουμε:

```
img2=fftshift(fft2(img1));
```

Για να απεικονίσουμε το πλάτος πρέπει να γράψουμε:

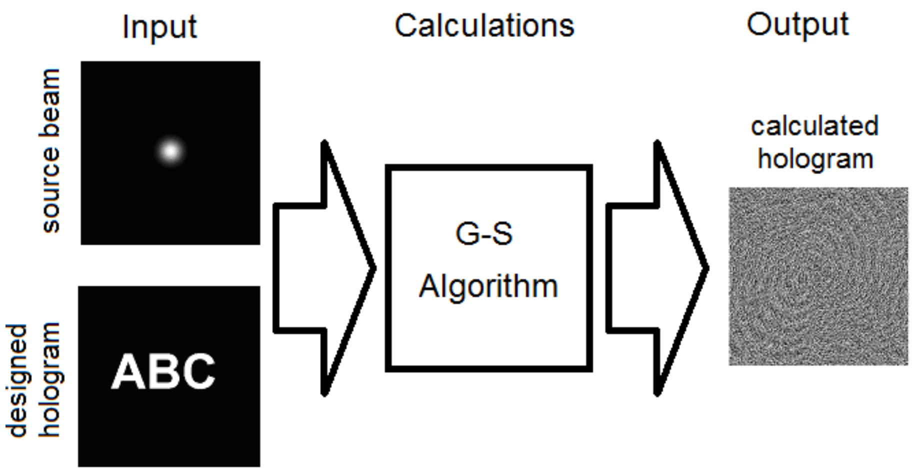
```
imshow(abs(img2),[]);
```

Ενώ για τη φάση πρέπει να γράψουμε:

```
imshow(angle(img2),[]);
```

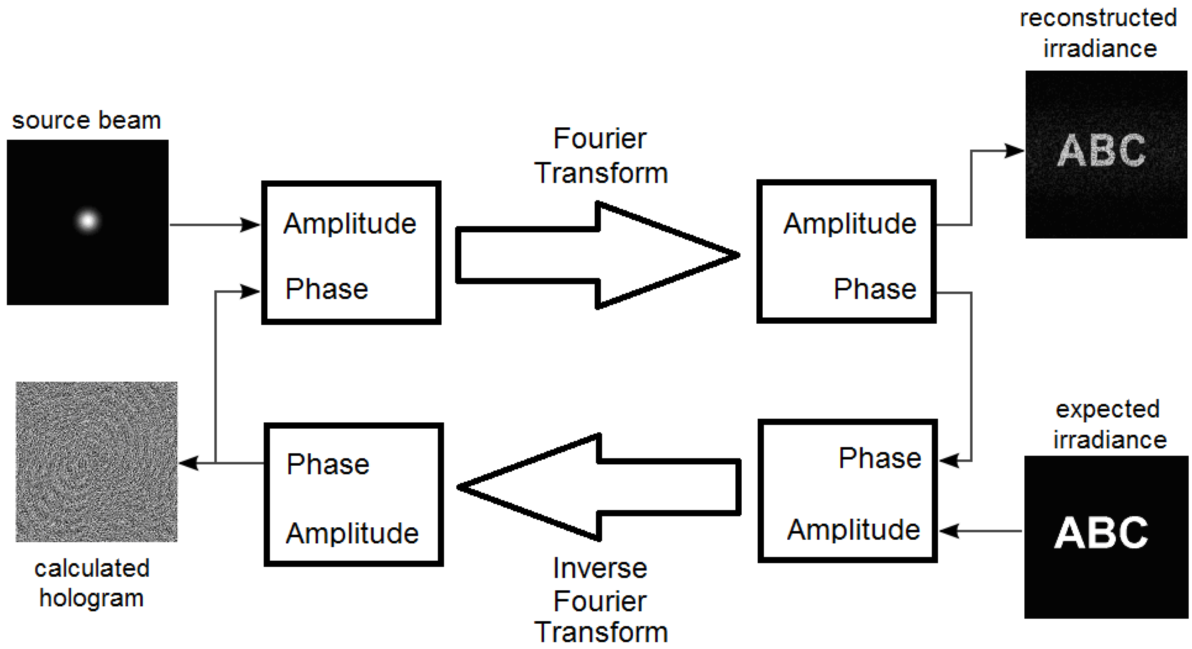
3.7. Αλγοριθμικό υπολογισμός των παραγόμενων μέσω υπολογιστή Ολογραμμάτων (CGH)

Για να δημιουργήσουμε ένα αναλογικό ολόγραμμα, χρειάζεται να έχουμε ένα αληθινό αντικείμενο, το οποίο θα εγγραφεί ως ολόγραμμα. Στην περίπτωση των παραγόμενων μέσω υπολογιστή ολογραμμάτων, το πραγματικό αντικείμενο αντικαθίσταται από ένα γραφικό αρχείο που περιέχει ένα οποιοδήποτε σχήμα (π.χ. ένα γραφικό αρχείο που περιέχει ένα λευκό κείμενο σε μαύρο υπόβαθρο). Η παραγωγή ενός ολογράμματος εμπλέκει τον υπολογισμό του μοτίβου συμβολής. Στη συνέχεια, το μοτίβο αυτό μπορεί να εγγραφεί σε μια ολογραφική επιφάνεια ή φιλμ. Το φως του λέιζερ που θα περάσει μια τέτοια επιφάνεια, θα σκεδαστεί με τέτοιο τρόπο που θα δημιουργήσει το αρχικό μοτίβο (π.χ. το κείμενο). Η παραγωγή του ολογράμματος με τη χρήση του υπολογιστή γίνεται χρησιμοποιώντας έναν αλγόριθμο που ονομάζεται Gerchberg-Saxton. Αυτός ο αλγόριθμος απαιτεί την εισαγωγή πληροφοριών σχετικά με την κατανομή της σχεδιασμένης έντασης (στο παράδειγμά μας, την προαναφερθείσα εικόνα με το κείμενο) και την κατανομή της έντασης της δέσμης φωτός με την οποία θα δημιουργηθεί το ολόγραμμα (που είναι επίσης στη μορφή ενός γραφικού αρχείου). Το αποτέλεσμα των αλγοριθμικών χειρισμών είναι η απόκτηση ενός γραφικού αρχείου με μια κατανομή φάσης που αποτελεί το ολόγραμμα. Η γενική ιδέα των παραπάνω απεικονίζεται στην εικόνα 61.



Εικ. 61 Τα εισαγόμενα δεδομένα βάσει των οποίων υπολογίζεται το ολόγραμμα

Ο αλγόριθμος Gerchberg-Saxton είναι ένας επαναληπτικός αλγόριθμος,. Συνεπώς κάποιοι χειρισμοί (υπολογισμοί) γίνονται πολλαπλές φορές. Κάθε εκτέλεση του υπολογισμού οδηγεί στο να αποκτήσουμε ένα ολόγραμμα ολοένα και καλύτερης ποιότητας. Η διαδικασία εφαρμογής του αλγόριθμου G-S παρουσιάζεται στην εικόνα 62.



Εικ. 62 Ο αλγόριθμος Gerchberg-Saxton

Παρακάτω είναι ένας κώδικας που θα χρησιμοποιήσουμε για την παραγωγή ολογραμμάτων. Το ολόγραμμα που θα παραχθεί θα σωθεί στο αρχείο CGH.bmp. Δείγματα αρχείων που μπορούν να χρησιμοποιηθούν ως εισαγόμενα (input) μπορούν να βρεθούν στο holomakers.eu.

```
pkg install image
pkg load image

GaussianBeam = imread('GaussianBeam.png');
GaussianBeam = rgb2gray(GaussianBeam);

StartPhase=zeros(1024,1024);

Source=fft2(ifft2(GaussianBeam));
StartPhase=fft2(ifft2(StartPhase));
Source = abs(Source).*exp(1i*angle(StartPhase));

TargetImg = imread('target.bmp');
Target=fft2(ifft2(TargetImg));

A = fftshift(ifft2(fftshift(Target)));
for i=1:20
  B = abs(Source).* exp(1i*angle(A));
  C = fftshift(fft2(fftshift(B)));
  D = abs(Target).* exp(1i*angle(C));
  A = fftshift(ifft2(fftshift(D)));
end
```

```
figure(1);imshow(GaussianBeam);title('Source Intensity');
figure(2);imshow(Target);title('Expected Intensity');
figure(3);imshow(angle(A),[]);title('Calculated Hologram');
figure(4);imshow(abs(C),[]);title('Reconstructed Intensity');

A=angle(A);
A=A-min(A(:));
A=A/max(A(:));

imwrite(A, 'CGH.bmp');
```