

ჟ. ხუნწარია, ვ. სტეპერმანი, ლ. ხუნწარია

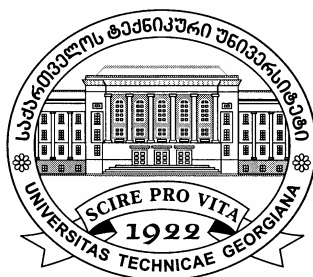
სიგნალების წყაროს
კოდირების პრინციპები

„ტექნიკური უნივერსიტეტი“

საქართველოს ტექნიკური უნივერსიტეტი

ჯ. ხუნწარია, ვ. სტეპერმანი, ლ. ხუნწარია

სიგნალების წყაროს კოდირების პრინციპები



დამტკიცებულია სტუ-ს
სარედაქციო-საგამომცემლო საბჭოს
მიერ. 02.07.2009, ოქმი №6

თბილისი
2009

სახელმძღვანელო "სიგნალების წყაროს კოდირების პრინციპები" გათვალისწინებულია ენერგეტიკისა და ტელეკომუნიკაციის ფაკულტეტის ტელეკომუნიკაციის სპეციალობის "ციფრული ტელეხედვის" მიმართულების ბაკალავრიატის სტუდენტებისათვის. იგი, აგრეთვე, გამოადგება ტელეკომუნიკაციის სპეციალობის სხვა მიმართულებების სტუდენტებს, მაგისტრანტებს და დოქტორანტებს, რომელთა ნაშრომების თემატიკაა სატელეკომუნიკაციო სიგნალების ციფრული დამუშავება და ეფექტური (ეკონომიური) კოდირება.

სახელმძღვანელოში გადმოცემულია ანალოგური სატელეკომუნიკაციო ერთგანზომილებიანი და ორგანზომილებიანი სიგნალების ციფრული სახით წარმოდგენის თავისებურებანი, დასაბუთებულია ციფრული სიგნალების ეკონომიური კოდირების აუცილებლობა მათში არსებული სტატისტიკური და ადამიანის ფსიქოფიზიოლოგიური თავისებურებების გათვალისწინების საფუძველზე. განხილულია სიგნალების კომპრესიის მეთოდები ინფორმაციის დანაკარგების გარეშე და დანაკარგებით. გადმოცემულია ხაფმენისა და არითმეტიკული კოდირებისა და დეკოდირების ალგორითმები, აგრეთვე გამოსახულებათა და მეტყველების სიგნალების ეფექტური კოდირებისათვის გამოყენებული მეთოდების არსი, მათი შედარებითი ანალიზი, ღირსებები და ნაკლოვანებები.

რეცენზენტი სრული პროფესორი ჯ. ბერიძე

© საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 2009

ISBN 978-9941-14-681-7

<http://www.gtu.ge/publishinghouse/>



ყველა უფლება დაცულია. ამ წიგნის არც ერთი ნაწილი (იქნება ეს ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) არანაირი ფორმით და საშუალებით (იქნება ეს ელექტრონული თუ მექანიკური), არ შეიძლება გამოყენებულ იქნას გამომცემლის წერილობითი ნებართვის გარეშე.

სავტორო უფლებების დარღვევა ისჯება კანონით.

ს ა რ ჩ ე ვ ი

	შესავალი	5
1.	ინფორმაციის გადაცემის საფუძვლები	7
1.1.	ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მოდელი	7
1.2.	ინფორმაციის წყარო	14
1.3.	დისკრეტიზაციის თეორემა	17
1.3.1.	რეალური სიგნალების დისკრეტიზაციის პრაქტიკული საკითხები	22
1.4.	ორგანზომილებიანი სიგნალების (გამოსახულებათა) დისკრეტიზაცია	24
1.5.	შეტყობინებათა დაკვანტვა. დაკვანტვის შეცდომები	27
1.6.	ინფორმაციის რაოდენობა. შეტყობინებათა წყაროს ენტროპია	32
1.6.1.	რთული შეტყობინების ენტროპია. წყაროს სიჭარბე	35
2.	ეკონომიური კოდირების საფუძვლები	41
2.1.	მონაცემთა კომპრესიის (შეკუმშვის) მიზანი და კომპრესიის სისტემების ტიპები	50
2.1.1.	კომპრესია ინფორმაციის დანაკარგის გარეშე	51
2.1.2.	კომპრესია ინფორმაციის დანაკარგებით	52
2.2.	კოდები მახსოვრობის გარეშე. ხაფმენის კოდები	57
2.2.1.	ხაფმენის ალგორითმი	59
2.2.2.	ენტროპიის საზღვრები ხაფმენის კოდისათვის	60
2.3.	კოდები მახსოვრობით	63
2.4.	ართმეტიკული კოდირება	65
2.4.1.	კოდირება	67
2.4.2.	დეკოდირება	69
2.5.	კოდირების ლექსიკონური მეთოდები. ზიგა-ლემპელის მეთოდი	72
2.5.1.	კოდირება	74
2.5.2.	დეკოდირება	75
2.6.	გამეორებათა სიგრძის კოდირება	77
2.7.	დიფერენციალური კოდირება	80
2.8.	კომპრესიის მეთოდები ინფორმაციის დანაკარგებით	81
2.8.1.	კოდირება გარდასახვებით. შეკუმშვის JPEG სტანდარტი	83
2.8.2.	ფრაქტალური მეთოდი	93
2.8.3.	რეკურსიული (ტალღური) ალგორითმი	96

2.9.	მოძრავ გამოსახულებათა (ვიდეო) კომპრესიის მეთოდები	98
2.10.	მეტყველების სიგნალების კომპრესიის მეთოდები	103
2.10.1.	სიგნალის ფორმის კოდირება	108
2.10.2.	წყაროს კოდირება	113
2.10.3	მეტყველების სიგნალების კოდირების ჰიბრიდული მეთოდები	119
დანართი 1.	შეტყობინებათა წყაროს ენტროპია, დაკვანტვა და დაკვანტვის შეცდომები. შესაბამისი ალგორითმების რეალიზაციის პროგრამა	127
დანართი 2.	დისკრეტული კოსინუსური გარდასახვის რეალიზაციის პროგრამა	129
დანართი 3.	რეკურსიული (ტაღღური) ალგორითმი რეალიზაციის პროგრამა	135
	ლიტერატურა	139

შ ე ს ა ვ ა ლ ი

ინფორმაციის მიღება, გადაცემა, დამუშავება და შენახვა – ეს ადამიანის მოღვაწეობის დინამიურად განვითარებადი და პერსპექტიული სფეროა უკანასკნელ ათწლეულში. მთელ მსოფლიოში ათეულ ათასობით მსხვილი, საშუალო და მცირე ფირმები, რომელთა წლიური ბრუნვა ასეული მილიარდობით დოლარია, დაკავებულია ინფორმაციის გადაცემის სხვადასხვა სისტემებისა და მოწყობილობების დამუშავებით, წარმოებით, გაყიდვითა და ექსპლუატაციით.

ინფორმაციის გადაცემისათვის განკუთვნილ სხვადასხვა სისტემებსა და გადაცემის მეთოდებში გარკვევა, მათი აგებისა და მუშაობის საერთო პრინციპების გაგება და ინფორმაციის გადაცემის სისტემების დამუშავებისათვის თანამედროვე მიდგომების შესწავლა წარმოადგენს წინამდებარე კურსის მიზანს.

უპირველეს ყოვლისა, განვსაზღვროთ შესწავლის საგანი.

ცნობილია, რომ *ინფორმაციის გადაცემის რადიოტექნიკური სისტემები* განკუთვნილია არა რაიმე ობიექტის, ენერჯის ან სიგნალების გადაცემისათვის, არამედ *ინფორმაციის* გადაცემისათვის. ამ სისტემებში ელექტრომაგნიტური ტალღები (სიგნალები) გამოიყენება მხოლოდ როგორც *ინფორმაციის გადამტანები წყაროდან მომხმარებელამდე*. ამაში მდგომარეობს მათი მთავარი განსხვავება სხვა მრავალ რადიოტექნიკურ სისტემასთან შედარებით, რომლებიც ასევე ახორციელებენ რადიოსიგნალების გადაცემასა და მიღებას. *ინფორმაციის გადაცემის რადიოტექნიკურ სისტემებს*, ანუ, სხვანაირად, *დამაკავშირებელი ტიპის რადიოტექნიკურ სისტემებს*, მიეკუთვნება რადიოკავშირის (მათ შორის ფიჭური რადიოსატელეფონო კავშირის), რადიომაუწყებლობისა და ტელეხედვის (მათ შორის თანამგზავრული), რადიოარხებით გაზომვების შედეგების გადაცემისათვის განკუთვნილი რადიოტელემეტრული სისტემები და სხვა.

გასაზომი ტიპის რადიოტექნიკური სისტემები, კერძოდ, რადიოლოკაციური და რადიონავიგაციური სისტემები, ტრაექტორული გაზომვის სისტემები, გარემოს პარამეტრების გაზომვისათვის განკუთვნილი სისტემები და სხვა, *ინფორმაციის გადაცემის რადიოტექნიკური სისტემებისაგან* განსხვავდებიან იმით, რომ *სასარგებლო ინფორმაცია ზედ ედება სიგნალს (ან აღიძვრება სიგნალში) გარემოსთან და ობიექტებთან ამ უკანასკნელის ურთიერთქმედების პროცესში* და ასახავს ამ ობიექტებისა და გავრცელების არის პარამეტრებსა და თვისებებს.

მიმღებ პუნქტში სასარგებლო ინფორმაცია გამოიყოფა სიგნალიდან ელექტრომაგნიტური ველის შესაბამისი პარამეტრების გაზომვის გზით.

რას წარმოადგენს ინფორმაცია და რაში მდგომარეობს მისი გადაცემის არსი? არსებობს ამის საკმაოდ მრავალი განსაზღვრება, თუმცა უკანასკნელ პერიოდამდე ყველაზე მეტად ადექვატურად ითვლებოდა *შენონისებური განსაზღვრება, რომლის მიხედვითაც ინფორმაცია გაურკვეველობის საზომია* (იმის დასახელების ხარისხი, რაც გადაცემისათვისაა განკუთვნილი). შესაბამისად, ინფორმაციის გადაცემის მიზანია მოცემული *გაურკვეველობის მოხსნა*.

ინფორმაციის ეს კლასიკური განსაზღვრება პირველად იქნა შემოტანილი კლოდ შენონის მიერ 1948 წელს დაწერილ შრომაში "კავშირგაბმულობის მათემატიკური თეორია", რამაც მრავალი ათწლეულის განმავლობაში განსაზღვრა ის მიდგომები ინფორმაციის გადაცემის მეთოდებისა და სისტემების ანალიზისა და დამუშავებისადმი, რომლებიც დღესაც უმთავრესად ამ განსაზღვრებაზეა ორიენტირებული. ამ მიდგომის შესაბამისად ინფორმაციის მიღების კვალობაზე *იხსნება გაურკვეველობა*. ამასთანავე, *რაც მეტი* ინფორმაცია მიიღება, *მით ნაკლებია* მიმღების გაურკვეველობის ხარისხი.

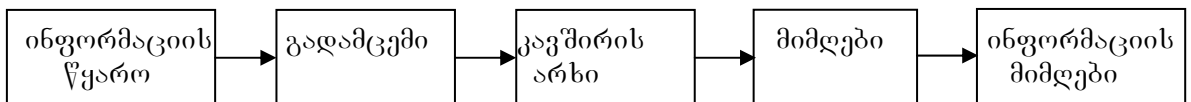
ინფორმაციის გადაცემის სისტემების თეორიისა და პრაქტიკის განვითარებამ დროთა განმავლობაში გამოკვეთა მთელი რიგი შეუსაბამობები აღნიშნულ განსაზღვრებასა და იმ შედეგებს შორის, რომლებიც ზოგჯერ გაცილებით უკეთესია არსებული თეორიის მიერ ნაწინასწარმეტყველებულთან შედარებით. მაგალითად, შენონისებური მიდგომა სულაც არ ითვალისწინებს *ინფორმაციის სარგებლიანობისა და მისი გაცნობიერების ხარისხს*, საგნის ან მოვლენის შესახებ *აპრიორული ცოდნის არსებობას და სხვა*. ეს მიდგომა მიმართულია *არა ცოდნის გაზრდისაკენ, არამედ – უცოდინრობის შემცირებისაკენ*, ანუ გარკვეული გაგებით ის პასიურია. ამასთან დაკავშირებით უკანასკნელი ორი ათწლეულის განმავლობაში მიმდინარეობს თითქოსდა ურყევი განსაზღვრებებისა და ინფორმაციის გადაცემის თეორიული საფუძვლების გადასინჯვა. მოცემული კურსის შესწავლის დროს განხილული იქნება როგორც კლასიკური მიდგომა, ასევე აღნიშნული საკითხის მიმართ არსებული თანამედროვე შეხედულება.

1. ინფორმაციის გადაცემის საშუალება

1.1. ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მოდელი

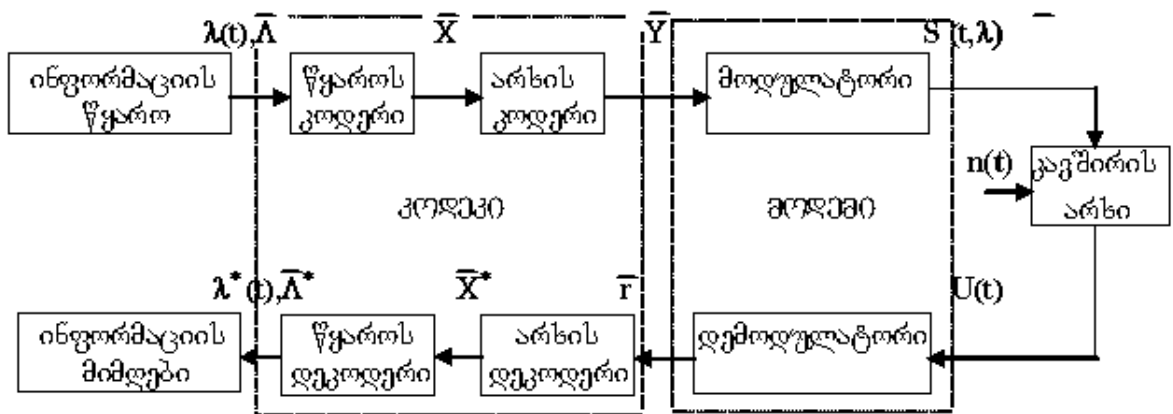
ინფორმაციის გადაცემის რადიოტექნიკური სისტემების შესწავლა დაიწყო საყოველთაოდ მიღებული მოდელების განხილვით, რომლებიც ნათელს ფენს მათი აგების საერთო პრინციპებსა და კანონზომიერებებს.

ზოგადი სახით ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მოდელი შეიძლება წარმოვადგინოთ შემდეგი სახით (ნახ. 1.1).



ნახ. 1.1. ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მოდელი

1.1 ნახაზზე წარმოდგენილი მოდელი თუმცა შეიცავს ინფორმაციის გადაცემის ნებისმიერი სისტემისათვის დამახასიათებელ ძირითად ელემენტებს, იგი მაინც წარმოადგენს ინფორმაციის გადაცემის რადიოტექნიკური სისტემის აღწერის უმარტივეს ილუსტრაციას, ვინაიდან პრაქტიკულად იგი არ ასახავს იმ მოქმედებებს, რომლებიც უნდა განხორციელდეს (ან შეიძლება განხორციელდეს) ინფორმაციის გადაცემისათვის წყაროდან მომხმარებელამდე. ამ თვალსაზრისით უფრო სრულყოფილია *ინფორმაციის გადაცემის (შენახვის) სისტემის* 1.2 ნახაზზე წარმოდგენილი მოდელი, რომლითაც ვისარგებლებთ მომავალში. უნდა აღინიშნოს, რომ სინამდვილეში ინფორმაციის გადაცემისას (თანაც არა მხოლოდ რადიოტალღების გამოყენებით) და ოპტიკურ დისკებზე, მაგნიტურ მატარებლებზე და კომპიუტერების მეხსიერებაში მისი შენახვისას წარმოქმნილი პრობლემები ერთმანეთს ძალიან გავს, რის გამოც ასევე იდენტურია მათი გადაწყვეტის მეთოდები და ტექნიკური მოწყობილობების სტრუქტურა.



ნახ. 1.2. ინფორმაციის გადაცემის (შენახვის) სისტემის მოდელი

მოკლედ დავახასიათოთ ამ მოდელის ელემენტების დანიშნულება და ფუნქციები.

1. ინფორმაციის ან შეტყობინების წყარო – ფიზიკური ობიექტი, სისტემა ან მოვლენაა, რომელიც აფორმირებს გადასაცემ შეტყობინებას. შეტყობინება კი – ეს არის ობიექტის (სისტემის ან მოვლენის) აღმწერი რაიმე ფიზიკური სიდიდის მნიშვნელობა ან ცვლილება. როგორც წესი, პირველადი შეტყობინებები (სიტყვა, მუსიკა, გამოსახულებები, გარემოს პარამეტრების გაზომვის შედეგები და სხვა) წარმოადგენენ არაელექტრული ბუნების დროით ფუნქციებს. კავშირის არხში გადაცემის მიზნით ეს შეტყობინება გარდაიქმნება ელექტრულ სიგნალად, რომლის ცვლილება დროში $\lambda(t)$ ასახავს გადასაცემ შეტყობინებას. განსაკუთრებით უკანასკნელ პერიოდში გადასაცემი შეტყობინებების მნიშვნელოვანი ნაწილი თავისი ბუნებით სიგნალებს კი არ წარმოადგენს, არამედ ისინი შეიძლება იყოს რიცხვების მასივები, ტექსტური ან სხვა სახის ფაილები და მათი მაგვარი რამ. ასეთი ტიპის შეტყობინებები შეიძლება წარმოვადგინოთ A ვექტორების სახით.

2. წყაროს კოდირი. საწყისი შეტყობინებების უდიდესი ნაწილი (სიტყვა, მუსიკა, გამოსახულებები და ა.შ.) განკუთვნილია ადამიანის შეგრძნების ორგანოების უშუალო აღქმისათვის და ზოგადად ცუდადაა მისადაგებული კავშირის არხების საშუალებით მათი ეფექტური გადაცემისადმი. ამიტომ, როგორც წესი, ხორციელდება შეტყობინებების $(\lambda(t)$ ან A) კოდირება. კოდირების პროცესში ჩვეულებრივ რთავენ უწყვეტი $\lambda(t)$ შეტყობინების დისკრეტიზაციასაც, ანუ მას წარმოადგენენ ელემენტარული დისკრეტული $\{\lambda_i\}$ შეტყობინებების მიმდევრობის სახით.

ზოგადად კოდირების ცნება გულისხმობს შეტყობინების $A\{\lambda_i\}$ ($i=1,2,\dots,K$) ალფაბიტის გარდასახვას (გარდაქმნას) გარკვეული წესით შერჩეული კოდური სიმბოლოების $\mathcal{R}\{x_j\}$ ($j=1,2,\dots,N$) ალფაბიტად. ჩვეულებრივ (მაგრამ არა აუცილებლად), კოდური სიმბოლოების ალფაბიტის ზომა $\dim \mathcal{R}\{x_j\}$ ნაკლებია ან მნიშვნელოვნად ნაკლებია წყაროს ალფაბიტის $\dim A\{\lambda_i\}$ ზომაზე. შეტყობინებათა კოდირებას შეიძლება ჰქონდეს სხვადასხვა მიზანი, კერძოდ – გადასაცემი მონაცემების მოცულობის შემცირება (მონაცემთა შეკუმშვა, კომპრესია), დროის ერთეულში გადასაცემი ინფორმაციის რაოდენობის გაზრდა, გადაცემის უტყუარობის ამაღლება, გადაცემისას საიდუმლოების უზრუნველყოფა და ა.შ.

ინფორმაციის გადაცემის რადიოტექნიკური სისტემის წყაროს კოდირების ცნებაში ვიგულისხმობთ ინფორმაციის მოცულობის შემცირება (შეკუმშვა, კომპრესია) მისი გადაცემის სიჩქარის ამაღლების ან მისი გადაცემისათვის საჭირო სიხშირეთა ზოლის შემცირების მიზნით.

წყაროს კოდირებას ზოგჯერ უწოდებენ ეკონომიურ, სიჭარბის გარეშე ან ეფექტურ კოდირებას, აგრეთვე მონაცემთა შეკუმშვას (კომპრესიას). მოცემულ შემთხვევაში ეფექტურობა გულისხმობს მონაცემთა მოცულობის შემცირების (შეკუმშვის) ხარისხს, რომელსაც უზრუნველყოფს კოდირების პროცესი.

თუ შეკუმშვა ხდება ისე, რომ შეკუმშული (კომპრესირებული) მონაცემებით შესაძლებელი იქნება საწყისი ინფორმაციის აღდგენა აბსოლუტურად ზუსტად, მაშინ კოდირებას ეწოდება არადამრღვევი. იგი გამოიყენება ტექსტური ინფორმაციის, ციფრული მონაცემების, კომპიუტერული ფაილების და ა.შ. გადაცემის ან შენახვისას, ანუ იქ, სადაც დაუშვებელია საწყის და აღდგენილ მონაცემებს შორის თუნდაც უმნიშვნელო განსხვავება.

ხშირად არაა აუცილებელი წყაროდან მომხმარებლისაკენ ინფორმაციის გადაცემა აბსოლუტური სიზუსტით, მითუმეტეს, რომ კავშირის არხში ყოველთვის არსებობს ხელშეშლები, რის გამოც აბსოლუტურად ზუსტი გადაცემა პრინციპში შეუძლებელია. ასეთ შემთხვევებში შეიძლება გამოყენებული იყოს დამრღვევი კომპრესია, რომელიც უზრუნველყოფს საწყისი შეტყობინების აღდგენას ამა თუ იმ ხარისხით შეკუმშვის შედეგად მიღებული და საწყისთან მიახლოებული სახით.

როგორც წესი, შეკუმშვის დარღვევადი მეთოდები გაცილებით ეფექტურია დაურღვევად მეთოდებთან შედარებით.

ამრიგად, წყაროს კოდირის გამოსასვლელზე გადასაცემი გადასაცემი $\lambda(t)$ ან A შეტყობინების შესაბამისად ფორმირდება კოდური სიმბოლოების X მიმღევრობა, რომელსაც ინფორმაციული მიმღევრობა ეწოდება და რომელიც რაც შეიძლება ნაკლები ზომის შემთხვევაშიც კი უზრუნველყოფს საწყისი შეტყობინების აბსოლუტურად ზუსტად (ან მიახლოებით) აღდგენას.

3. არხის კოდირი. კავშირის ხელშეშლებიანი არხით ინფორმაციის გადაცემისას მიმღებში მონაცემები შეიძლება მიღებულ იქნას შეცდომებით. თუ ასეთ შეცდომებს გააჩნია მცირე სიდიდე და წარმოიქმნებიან საკმაოდ იშვიათად, მაშინ ინფორმაცია შეიძლება გამოიყენოს მომხმარებელმა. შეცდომათა დიდი რაოდენობის შემთხვევაში კი მიღებული ინფორმაციის გამოყენება არ შეიძლება.

კოდირება არხში, ანუ ხელშეშლამდგრადი კოდირება, წარმოადგენს გადასაცემი მონაცემების დამუშავების მეთოდს, რომელიც უზრუნველყოფს იმ შეცდომების რაოდენობის შემცირებას, რომლებიც წარმოიქმნება ხელშეშლებიან არხებში გადაცემის პროცესში. არსებობს ინფორმაციის ხელშეშლამდგრადი კოდირების მრავალი სხვადასხვა მეთოდი, თუმცა თითოეული მათგანის შემთხვევაში გადასაცემ შეტყობინებაში შეყავთ სპეციალურად შექმნილი სიჭარბე (გადასაცემ კოდურ მიმღევრობებს უმატებენ ჭარბ სიმბოლოებს), რომელიც მიმღებ მხარეს იძლევა წარმოქმნილი შეცდომების აღმოჩენისა და გასწორების საშუალებას. ამრიგად, თუ წყაროს კოდირებისას ხდება შეტყობინებაში არსებული ბუნებრივი სიჭარბის მოცილება ან შემცირება, არხში კოდირებისას – გადასაცემ შეტყობინებაში შეგნებულად შეაქვთ სიჭარბე. ამის შედეგად არხის კოდერის გამოსასვლელზე ფორმირდება კოდური სიმბოლოების $Y(X)$ მიმღევრობა, რომელსაც კოდურ მიმღევრობას უწოდებენ.

უნდა აღინიშნოს, რომ ინფორმაციის გადაცემისას როგორც ხელშეშლამდგრადი კოდირება, ასევე მონაცემთა შეკუმშვა (კომპრესია) არაა აუცილებლად შესასრულებელი ოპერაციები. ეს პროცედურები და მათი შესაბამისი ბლოკები ინფორმაციის გადაცემის რადიოტექნიკური სისტემის სტრუქტურულ სქემაში შეიძლება არ არსებობდეს. თუმცა ამან შეიძლება გამოიწვიოს სისტემის ხელშეშლამდგრადობის მნიშვნელოვანი დანაკარგები, გადაცემის სიჩქარის მნიშვნელოვანი შემცირება და ინფორმაციის გადაცემის ხარისხის გაუარესება. ამიტომ პრაქტიკულად ყველა თანამედროვე სისტემა (უმარტივესი სისტემების გარდა) უნდა შეიცავდეს და აუცილებლად შეიცავს კიდევ მონაცემთა როგორც ეფექტურ, ასევე ხელშეშლამდგრად კოდირებას.

4. მოდულატორი. ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მოდულატორის ფუნქციებია წყაროს შეტყობინების ან კოდერის მიერ გამომუშავებული კოდური მიმღევრობების შეთანხმება კავშირის არხის თავისებურებებთან და კავშირის საერთო არხით (როგორცაა რადიოარხი) დიდი რაოდენობის შეტყობინების გადაცემის შესაძლებლობის უზრუნველყოფა.

მართლაც, უწყვეტი $\lambda(t)$ და დისკრეტული A გადასაცემი შეტყობინებებიდან უმრავლესი მათგანი და აგრეთვე მათი კოდირების შედეგები – კოდური სიმბოლოების X და Y მიმღევრობები წარმოადგენენ შედარებით დაბალსიხშირულ სიგნალებს შედარებით ფართო ზოლით ($\Delta F \ll f_0$ მშც, $\Delta F \sim f_0$). ამავე დროს ეფექტური გადაცემა ელექტრომაგნიტური ტალღების (რადიოტალღების) გამოყენებით შესაძლებელია მხოლოდ საკმაოდ

მაღალსიხშირული სიგნალებისათვის ($f_0 \geq 1 \dots 1000$ მჰც და უფრო ზევით), რომლებსაც გააჩნია შედარებით ვიწროხოლოვანი სპექტრები ($\Delta F \ll f_0$). ამიტომ მოდულატორმა წყაროს $\lambda(t)$ (A) შეტყობინებები ან მათი შესაბამისი კოდური X და Y მიმღევრობები უნდა გარდაქმნას $S(t, \lambda(t))$, $S(t, Y(\lambda(t)))$ სიგნალებად (მოახდინოს შეტყობინებათა ზედდება სიგნალებზე), რომელთა თავისებურებები შესაძლებელს გახდიდა რადიოარხების საშუალებით (ან სხვა არსებული – სატელეფონო, ოპტიკური და ა.შ. არხებით) მათ ეფექტურ გადაცემას. ამასთან ინფორმაციის გადაცემის იმ მრავალი სისტემის შესაბამისი სიგნალები, რომლებიც მუშაობენ საერთო რადიოარხში, ისეთი უნდა იყოს, რომ უზრუნველყოფილი იყოს ყველა წყაროდან მიღებული შეტყობინების დამოუკიდებელი გადაცემა ინფორმაციის ყველა მიმღებამდე.

დღეისათვის არსებობს სიგნალების მოდულაციის მრავალი მეთოდი, რომლებსაც განსხვავებული ეფექტურობა გააჩნიათ და რომლებიც უზრუნველყოფენ ინფორმაციის გადაცემას ამა თუ იმ ხარისხით. მათ შორის ყველაზე მარტივია უწყვეტი სიგნალების ამპლიტუდური, სიხშირული და ფაზური მოდულაციები. წინამდებარე კურსში განხილული იქნება აგრეთვე ინფორმაციის გადაცემის რადიოტექნიკურ სისტემებში გამოყენებული სიგნალების მოდულაციის უფრო თანამედროვე და უფრო ეფექტური მეთოდები, მათ შორის ისეთები, რომლებიც გამოიყენებენ ფართოხოლოვან ხმაურის მსგავს სიგნალებს. ამასთან, მოდულაციის პროცედურა განხილული იქნება არა უბრალოდ $S(t)$ სიგნალის პარამეტრების ცვლილება გადასაცემი $\lambda(t)$ შეტყობინების მნიშვნელობათა მიხედვით, არამედ როგორც შეტყობინების გარდაქმნა სიგნალად.

5. კავშირის არხი. განმარტების თანახმად, ინფორმაციის გადაცემის რადიოტექნიკური სისტემა ინფორმაციის გადაცემის სისტემაა, რომელიც წყაროდან მომხმარებლისაკენ ინფორმაციის გადამტანად იყენებს ელექტრომაგნიტურ ტალღებს ან რადიოტალღებს, ხოლო გავრცელების არედი – გარემომცველ სივრცეს ანუ რადიოარხს. ამაში მდგომარეობს ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მთავარი განსხვავება ინფორმაციის გადაცემის სხვა სისტემებთან შედარებით, რომლებიც იყენებენ სადენიან, ბოჭკოვან-ოპტიკურ, აკუსტიკურ და სხვა მსგავს არხებს. სხვა მხრივ, არაარსებითი დეტალების მხედველებაში მიუღებლად, ასეთი სისტემების სტრუქტურა და ძირითადი ელემენტების ფუნქციები იდენტურია.

რადიოარხის შეიძლება განვიხილოთ როგორც ინფორმაციის გადაცემის რადიოტექნიკური სისტემის ერთ-ერთი რგოლი, რომლის შესასვლელს მიეწოდება გადაცემის $S(t, \lambda(t))$ სიგნალი, ხოლო გამოსასვლელზე მიიღება $U(t)$ სიგნალი, რომელსაც ჩვეულებრივ მიღებულ რხევას უწოდებენ.

არსებობს დიდი ან მცირე სირთულის რადიოარხის მრავალი მოდელი, თუმცა ზოგადად სიგნალი $S(t, Y(\lambda(t)))$, გაივლის რა კავშირის არხს, მიიღება (სუსტდება), დროში ყოვნდება (ფაზური ძვრა) და ხმაურიანდება. ამ შემთხვევაში მიღებული $U(t)$ რხევას აქვს შემდეგი სახე.

$$U(t) = \varepsilon S(t - \tau, Y(\lambda(t))) + n(t), \quad (1.1)$$

სადაც ε - მიღებაა, τ - დროითი დაყოვნებაა, ხოლო $n(t)$ - კავშირის არხში არსებული ხმაურებია.

მიმღები. ინფორმაციის გადაცემის რადიოტექნიკური სისტემის მიმღების დანიშნულებაა მიღებული $U(t)$ რხევიდან თავის გამოსასვლელზე მაქსიმალურად შესაძლო სიზუსტით აღადგინოს გადაცემული $\lambda(t)$ შეტყობინება (ან A). ხელშემშლელის არსებობის გამო მიღებული (აღდგენილი, აღწარმოებული) შეტყობინება ზოგადად განსხვავდება გაგზავნილი შეტყობინებისაგან. მიღებულ შეტყობინებას ვუწოდოთ შეფასება (მხედველობაშია შეტყობინების შეფასება) და ავლნიშნოთ გაგზავნილი შეტყობინების აღმნიშვნელი სიმბოლოთი, ოღონდ დავეუმატოთ ნიშანი *, ანუ მისი აღნიშვნა იქნება $\lambda^*(t)$ ან A^* . მიღებული რხევიდან შეტყობინების შეფასების აღდგენის (აღწარმოების) პროცესი საზოგადოდ მოიცავს რამდენიმე ეტაპს.

6. დემოდულატორი. შეტყობინების $\lambda^*(t)$ (ან A^*) შეფასების აღდგენისათვის სისტემის მიმღებმა, უპირველეს ყოვლისა, მიღებული $U(t)$ რხევისა და გადაცემისას გამოყენებული სიგნალის სახისა და მოდულაციის მეთოდის შესახებ მონაცემთა გათვალისწინებით უნდა მოახდინოს იმ კოდური $Y^*(\lambda(t))$ მიმდევრობის შეფასება, რომელსაც მიღებულ r მიმდევრობას უწოდებენ. ამ პროცედურას დემოდულაციას, დეტექტირებას ან სიგნალის მიღებას უწოდებენ. ამასთან დემოდულაცია უნდა ისეთნაირად განხორციელდეს, რომ მიღებული r მიმდევრობა მინიმალურად განსხვავდებოდეს გადაცემული კოდური Y მიმდევრობისაგან. გადაწყვეტის მეთოდის მიხედვით მიღებული $U(t)$ მიმდევრობის დემოდულაციის ამოცანა ძირითადად ემთხვევა ხელშემშლელის ფონზე სიგნალის ოპტიმალური მიღების ამოცანის სხვადასხვა ვარიანტებს (ოპტიმალური აღმოჩენა, ორი ან რამდენიმე სიგნალის ოპტიმალური გარჩევითობა და ა.შ.).

7. არხის დეკოდერი. ზოგადად მიღებული r მიმდევრობები შეცდომების არსებობის გამო შეიძლება განსხვავდებოდნენ გადაცემული კოდური Y სიტყვებისაგან. ასეთი შეცდომების რაოდენობა დამოკიდებულია კავშირის არხში ხელშეშლების დონეზე, სიგნალის გადაცემისათვის შერჩეულ გადაცემის სიჩქარესა და მოდულაციის მეთოდზე და, აგრეთვე, $U(t)$ რხევის მიღების (დემოდულაციის) მეთოდზე. *არხის დეკოდერის ამოცანაა* აღნიშნული შეცდომების აღმოჩენა და შეძლებისდაგვარად მათი *გასწორება*. მიღებულ r მიმდევრობაში შეცდომათა აღმოჩენისა და გასწორების პროცედურას *არხის დეკოდირებას* უწოდებენ. r -ის დეკოდირების შედეგია ინფორმაციული მიმდევრობის X^* შეფასება. ხელშეშლამდგრადი კოდისა და კოდირებისა და დეკოდირების მეთოდების შერჩევა უნდა განხორციელდეს ისე, რომ რაც შეიძლება ნაკლები რაოდენობის გაუსწორებელი შეცდომა აღმოჩნდეს არხის დეკოდერის გამოსასვლელზე.

ინფორმაციის გადაცემის (შენახვის) სისტემებში ხელშეშლამდგრადი კოდირება/დეკოდირების საკითხებს დღეისათვის ეთმობა განსაკუთრებული ყურადღება, ვინაიდან ასეთი მიდგომა იძლევა ინფორმაციის გადაცემის ხარისხის მნიშვნელოვანი გაუმჯობესების საშუალებას. ხშირ შემთხვევაში, როდესაც მიღებული ინფორმაციის უტყუარობისადმი წაყენებული მოთხოვნები ძალიან მკაცრია (მაგალითად, მონაცემთა გადაცემის კომპიუტერული ქსელებში, მართვის დისტანციურ სისტემებში და სხვა), გადაცემა ხელშეშლამდგრადი კოდირების გარეშე საერთოდ შეუძლებელია.

8. წყაროს დეკოდერი. ვინაიდან გადაცემის პროცესში წყაროს ($\lambda(t)$, A) ინფორმაციამ მისი უფრო კომპაქტურად (ან უფრო მოხერხებულად) წარმოდგენის მიზნით განიცადა კოდირება (*მონაცემთა შეკუმშვა, ეკონომიური კოდირება, წყაროს კოდირება*), ამიტომ მიმღებში მიღებული X^* მიმდევრობიდან საჭიროა ინფორმაციის საწყისი (ან თითქმის საწყისი) სახით აღდგენა. X^* -ის საშუალებით A^* -ს აღდგენის პროცედურას *წყაროს დეკოდირება* ეწოდება და ის შეიძლება იყოს ან უბრალოდ კოდირების შებრუნებული პროცედურა (არადამრღვევი კოდირება/დეკოდირება), ან კიდევ A^* -ს აღდგენა მოხდეს მიახლოებულად, რომელიც ბევრად ან ნაკლებად იქნება განსხვავებული A -საგან (დამრღვევი კოდირება/დეკოდირება). X^* -ის მიხედვით A^* -ს აღდგენის ოპერაციას აუცილებლობის შემთხვევაში მივაკუთვნოთ აგრეთვე უწყვეტი $\lambda^*(t)$ ფუნქციის აღდგენა A^* -ს დისკრეტული მნიშვნელობების ნაკრების მიხედვით.

უნდა აღინიშნოს, რომ უკანასკნელ დროს ინფორმაციის გადაცემის სისტემებში ეკონომიური კოდირების განხორციელებას სულ უფრო თვალსაჩინო ადგილი უჭირავს, ვინაიდან იგი ხელშეშლამდგრად კოდირებასთან ერთად აღმოჩნდა გადაცემის სიჩქარის გაზრდისა და ხარისხის გაუმჯობესების ყველაზე ეფექტური ხერხი.

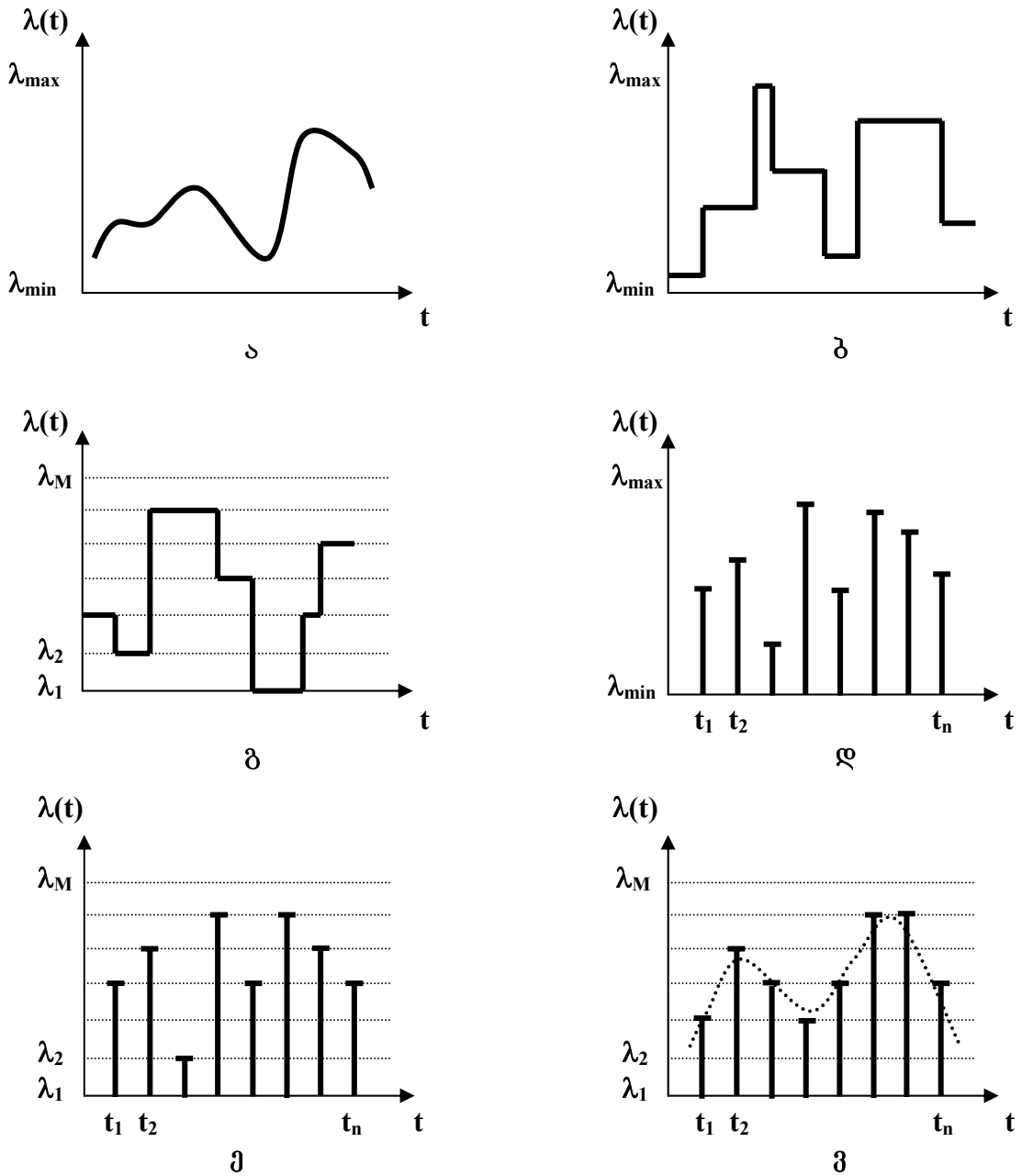
უფრო დეტალურად განვიხილოთ ინფორმაციის გადაცემის რადიოტექნიკური სისტემის ძირითადი ელემენტები, რომელთაგან პირველია *ინფორმაციის წყარო*.

1.2. ინფორმაციის წყარო

ინფორმაციის ან შეტყობინების წყარო გადასაცემი შეტყობინების მაფორმირებელი ობიექტი, სისტემა ან მოვლენაა, ხოლო შეტყობინება ობიექტის (სისტემის ან მოვლენის) ამსახველი რაიმე ფიზიკური სიდიდის მნიშვნელობა ან ცვლილებაა. როგორც წესი, პირველადი შეტყობინებები (სიტყვა, მუსიკა, გამოსახულებები, გარემომცველი სამყაროს პარამეტრების გაზომვები და ა.შ.) წარმოადგენენ არაელექტრული ბუნების (აკუსტიკური რხევა, ტემპერატურა, რაიმე სიბრტყეზე სიკაშკაშის განაწილება და სხვა) დროის $\lambda(t)$, ან სხვა არგუმენტების $\lambda(x,y,z)$ ფუნქციებს. კავშირის არხით გადაცემის მიზნით ეს შეტყობინებები ჩვეულებრივ გარდაიქმნებიან ელექტრულ სიგნალად, რომლის ცვლილებაც დროში $\lambda(t)$ ასახავს გადასაცემ ინფორმაციას. ასეთ შეტყობინებებსა და სიგნალებს უწოდებენ *უწყვეტ*, ან *ანალოგურ შეტყობინებებსა და სიგნალებს* და მათთვის სრულდება შემდეგი პირობა.

$$\lambda \in (\lambda_{min}, \lambda_{max}), t \in (0, t), \quad (1.2)$$

ანუ ასეთი შეტყობინებებისა და სიგნალებისათვის როგორც ფუნქციის, ასევე არგუმენტის მნიშვნელობა უწყვეტია ან განსაზღვრულია უწყვეტი ინტერვალის ნებისმიერი მნიშვნელობისათვის როგორც λ -ს, ასევე t -ს მიხედვით (ნახ 1.3,ა,ბ).



ნახ. 13. შეტყობინებებისა და სიგნალების სახეობები

მრავალი შეტყობინება, კერძოდ კი ისეთები, როგორიცაა შემსრულებელი მოწყობილობებისადმი გაცემული ბრძანებები, სატელეგრაფო შეტყობინებები, ტექსტური ინფორმაცია და სხვა, დისკრეტული ხასიათის მატარებელია. ამასთან ან $A(\lambda_i)$ შეტყობინების ალფაბეტი წარმოადგენს სასრულო თვლად

$$\lambda_i = \lambda_1, \lambda_2, \dots, \lambda_k, \quad i=1, K \quad (1.3)$$

სიმრავლეს (დისკრეტული ანუ ან მნიშვნელობათა მიხედვით დაკავანტული შეტყობინებები, ნახ. 13,გ), ან სიგნალები გადაიცემა დროის დისკრეტულ

$$t = t_1, t_2, \dots, t_m, \quad i=1, M \quad (1.4)$$

მომენტებში (დროში დისკრეტული შეტყობინებები, ნახ. 1.3,დ), ან კიდევ ერთიცა და მეორეც (დროში და მნიშვნელობათა მიხედვით დისკრეტული სიგნალები, ანუ, როგორც მათ უწოდებენ სხვანაირად ციფრული სიგნალები ან შეტყობინებები, ნახ.1.3,ე.ე).

განსაკუთრებით უკანასკნელ პერიოდში გადასაცემი შეტყობინებებიდან მრავალი მათგანი (მონაცემთა პაკეტები,სხვადასხვა პარამეტრების ციფრული გაზომვების შედეგები, ციფრული ფოტოსურათები, ტექსტური, გრაფიკული ან სხვა სახის ფაილები და სხვა) თავისი ბუნებით არ წარმოადგენს სიგნალს. ასეთი ტიპის შეტყობინებები შეიძლება წარმოვადგინოთ რიცხვების მასივების ან **A** ვექტორების სახით.

როგორც ზემოთ მოყვანილი მაგალითებიდან ჩანს, გადაცემისათვის განკუთვნილი შეტყობინებების (ან მათი ამსახველი სიგნალების) მრავალნაირობის მიუხედავად მათი უმრავლესობა შეიძლება მიეკუთვნოს მხოლოდ რამდენიმე ერთმანეთისაგან არსებითად განსხვავებულ სახეს, კერძოდ:

1. დროში უწყვეტი (ანალოგური) შეტყობინებები (სიგნალები).
2. დროის მიხედვით დისკრეტული (დისკრეტიზირებული) შეტყობინებები (სიგნალები).
3. დონის (მნიშვნელობათა) მიხედვით დისკრეტული (დაკვანტული) შეტყობინებები (სიგნალები).

როგორც ჩანს, ერთი შეხედვით ასეთ ძალზე განსხვავებული სიგნალებს, როგორიცაა უწყვეტი და დისკრეტიზირებული სიგნალები (იხ. ნახ. 1.3), გააჩნიათ მრავალი საერთო და ისინი ერთმანეთთან დაკავსირებულია ფუნქციონალური დამოკიდებულებით, რომელსაც ამყარებს თეორემა დისკრეტიზაციის შესახებ.

13. დისკრეტიზაციის თეორემა

ტელეკომუნიკაციის თეორიის განსაკუთრებულად მნიშვნელოვან ნაწილს წარმოადგენს ეგრეთ წოდებული თეორემა ანათვლების შესახებ. იგი იძლევა რეალურ ინფორმაციულ სიგნალებს შორის უმრავლეს *უწყვეტ სიგნალებსა* (სიტყვა, მუსიკა, სატელევიზიო გამოსახულებების აღმწერი ელექტრული სიგნალები, სხვადასხვა რადიოტექნიკური სისტემების წრედების სიგნალები და სხვა) და *დროის ცალკეულ მომენტებში ამ სიგნალების მნიშვნელობებს (ეგრეთ წოდებულ ანათვლებს) შორის*. აღნიშნული კავშირის გამოყენებაზე აიგება თანამედროვე ციფრული რადიოტექნიკა, კერძოდ, ბგერითი და სატელევიზიო სიგნალების გადაცემისა და შენახვის ციფრული მეთოდები, სატელეფონო და ფიჭური კავშირის ციფრული სისტემები, ციფრული თანამგზავრული ტელეხედვის სისტემები, და ა.შ. შეიძლება ითქვას უფრო მეტიც: სიგნალების დამუშავების ტექნიკის მომავალი მის ციფრულ რეალიზაციაშია. გაივლის კიდევ 10–20 წელი და ჩვენ მხოლოდ მოვიგონებთ სიგნალების ფორმირების, მიღების, დამუშავებისა და შენახვისათვის გამოყენებულ ტრადიციულ ანალოგურ მეთოდებს. ინფორმაციული სიგნალების დამუშავებასთან დაკავშირებული მთელი პრაქტიკული რადიოტექნიკა გადავა ციფრულ რეალიზაციებზე.

დისკრეტიზაციის თეორემა, ანუ, როგორც მას კიდევ უწოდებენ კოტელნიკოვის თეორემა ან უიტეკერის თეორემა, ფორმირდება შემდეგი სახით: *შეზღუდული სპექტრის მქონე უწყვეტი $X(t)$ ფუნქცია, ანუ ფუნქცია, რომელსაც თავის*

$$F\{X(t)\} = \int_{-\infty}^{\infty} X(t) e^{-j2\pi ft} dt \quad (1.5)$$

სპექტრში არ გააჩნია ისეთი სიხშირის შემდგენები, რომლებიც მოთავსებულია $f \in (-F_m, F_m)$ ზოლის გარეთ, მთლიანად განისაზღვრება $\Delta t \ll 1/F_m$ ინტერვალით (დისკრეტიზაციის პერიოდით) დაშორებულ დროის დისკრეტულ t_i მომენტებში ფორმირებული (აღებული) $X(t_i)$ ანათვლების მიმდევრობით.

აღნიშნული თეორემის დამტკიცება ემყარება სიგნალებსა და მათ სპექტრებს შორის ცალსახა შესაბამისობას. სხვა სიტყვებით რომ ვთქვათ, თუ სიგნალები ერთნაირია, მაშინ მათი შესაბამისი სპექტრებიც იქნება ერთნაირი და, პირიქით, თუ სპექტრები ერთნაირია, მაშინ სიგნალებიც ერთნაირი იქნება.

განვიხილოთ დისკრეტიზაციის თეორემის უმარტივესი დამტკიცება, რისთვისაც თავდაპირველად ვაჩვენოთ, თუ როგორაა დაკავშირებული $\{X(t_i)\}$

ანათვლების დისკრეტული მიმდევრობის სპექტრი უწყვეტი $X(t)$ ფუნქციის სპექტრთან.

უწყვეტი $X(t)$ ფუნქციის ანათვლების მიმდევრობა შეიძლება წარმოვადგინოთ $X(t)$ -სა და Δt პერიოდის მქონე δ -იმპულსების პერიოდული მიმდევრობის ნამრავლის სახით:

$$X(t_i) = X(t) \sum_{-\infty}^{\infty} \delta(t - i\Delta t). \quad (1.6)$$

მაშინ დისკრეტიზირებული $X(t_i)$ ფუნქციის სპექტრი (ფურიეს გარდასახვა) შეიძლება ჩაიწეროს შემდეგი სახით:

$$F\{X(t_i)\} = \int_{-\infty}^{\infty} X(t) \sum_{i=-\infty}^{\infty} \delta(t - i\Delta t) e^{-j2\pi f t} dt = \sum_{i=-\infty}^{\infty} \int_{-\infty}^{\infty} X(t) e^{-j2\pi f t} \delta(t - i\Delta t) dt, \quad (1.7)$$

ანუ δ -ფუნქციის "ფილტრაციის" თვისების გათვალისწინებით (1.7) ფორმულა მიიღებს საბოლოო სახეს:

$$F\{X(t_i)\} = \sum_{i=-\infty}^{\infty} X(i\Delta t) e^{-j2\pi f i \Delta t}. \quad (1.8)$$

არაა ძნელი იმის შემჩნევა, რომ პერიოდულად დისკრეტიზირებული $X(i\Delta t)$ ფუნქციის სპექტრი ასევე პერიოდულია, რომლის პერიოდია $1/\Delta t$.

მართლაც,

$$X_{\Delta}(f + m/\Delta t) = \sum_{i=-\infty}^{\infty} X(i\Delta t) e^{-j2\pi(f + m/\Delta t)i\Delta t} = \sum_{i=-\infty}^{\infty} X(i\Delta t) e^{-j2\pi f i \Delta t} e^{-j2\pi m i} = X_{\Delta}(f). \quad (1.9)$$

ასეთივე შედეგი, ოღონდ განსხვავებული ხერხით, შეიძლება მივიღოთ, თუ გავიხსენებთ, რომ დროით არეში ფუნქციების გადამრავლებას შეესაბამება მათი სპექტრების ხვევა. აღნიშნულის გათვალისწინებით

$$F\{X(t_i)\} = F\left\{X(t) \sum_{i=-\infty}^{\infty} \delta(t - i\Delta t)\right\} = F\{X(t)\} \otimes F\left\{\sum_{i=-\infty}^{\infty} \delta(t - i\Delta t)\right\}. \quad (1.10)$$

"მესერისებური ფუნქციის" სპექტრს ასევე აქვს δ -იმპულსების პერიოდული მიმდევრობის სახე, რომლის სიხშირეა $\Delta f = 1/\Delta t$, ანუ

$$F\left\{\sum_{i=-\infty}^{\infty} \delta(t - i\Delta t)\right\} = (1/\Delta t) \sum_{k=-\infty}^{\infty} \delta(f - k/\Delta t). \quad (1.11)$$

თუ გავითვალისწინებთ δ -ფუნქციის "ფილტრაციის" თვისებას, მაშინ ხვევის განხორციელების შედეგად მივიღებთ:

$$F\{X(t_i)\} = (1/\Delta t) \sum_{k=-\infty}^{\infty} F_{X(t)}(f-k/\Delta t). \quad (1.12)$$

ამრიგად, დისკრეტიზირებული $X(i, \Delta t)$ ფუნქციის სპექტრი მიიღება საწყისი $X(t)$ ფუნქციის სპექტრის $1/\Delta t$ პერიოდით გამეორების გზით.

(1.12) ფორმულიდან ცხადია აგრეთვე, რომ $k=0$ -თვის

$$F\{X(t_i)\}_{k=0} = (1/\Delta t) F\{X(t)\}, \quad (1.13)$$

ანუ დისკრეტიზირებული ფუნქციის სპექტრის შემდგენი $k=0$ -თვის მუდმივი $1/\Delta t$ მამრავლის სიზუსტით ემთხვევა საწყისი უწყვეტი $X(t)$ ფუნქციის სპექტრს.

ამრიგად, თუ $X(t_i)$ მიმდევრობის სრული (პერიოდული) სპექტრიდან რაიმე საშუალებით გამოვეყოფთ მხოლოდ შემდგენს, რომლისთვისაც $k=0$, მაშინ დისკრეტული $X(t_i)$ მიმდევრობიდან სრულად აღდგება საწყისი უწყვეტი $X(t)$ ფუნქცია.

(1.13) ფორმულიდან გამომდინარეობს, რომ $X(t_i)$ მიმდევრობის სპექტრიდან საწყისი $X(t)$ სიგნალის სპექტრის გამოყოფა შესაძლებელია იდეალური დაბალი სიხშირეების ფილტრით, რომლის სიხშირული მახასიათებელია

$$H(f) = \begin{cases} \Delta t & |f| \leq 1/2\Delta t, \\ 0, & |f| > 1/2\Delta t. \end{cases} \quad (1.14)$$

ამასთანავე სპექტრის სხვადასხვა k -ს შესაბამისი შემდგენები შეიძლება განცალკევდნენ მათი ურთიერთგადაუფარაობის შემთხვევაში. სპექტრის შემდგენების გადაუფარაობა კი უზრუნველყოფილი იქნება შემდეგი პირობის შესრულებისას:

$$F_m \geq (1/\Delta t), \quad \text{ანუ} \quad \Delta t \leq 1/F_m, \quad (1.15)$$

საიდანაც გამომდინარეობს დისკრეტიზაციის Δt ინტერვალის (პერიოდის) ის მნიშვნელობა, რომელიც უზრუნველყოფს საწყისი $X(t)$ სიგნალის აღდგენას მისი ანათვლების მიმდევრობის საშუალებით.

ანათვლების დისკრეტული მიმდევრობიდან უწყვეტი სიგნალის აღმდგენი ფილტრის იმპულსური მახასიათებელი შეიძლება მიღოს (1.14) ფორმულით წარმოდგენილი მისი სიხშირული მახასიათებლის ფურიეს გარდაქმნით და მას აქვს შემდეგი სახე:

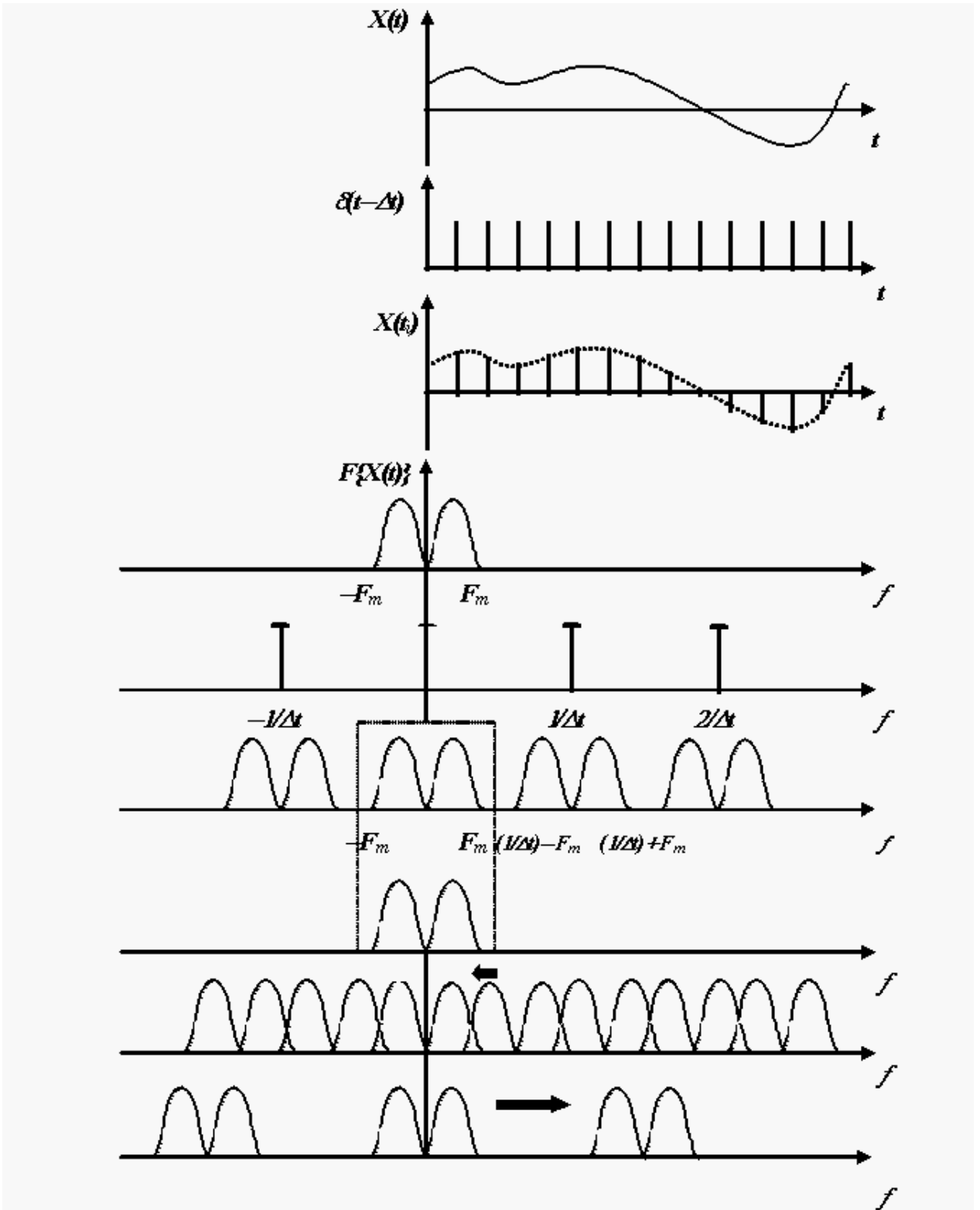
$$h(\tau) = F^{-1}\{H(f)\} = \text{sinc}(2\pi F_m \tau). \quad (1.16)$$

თუ $X(t)$ იმპულსურ მიმდევრობას გავატარებთ $h(\tau)$ იმპულსური მახასიათებლის მქონე ფილტრში, მაშინ მივიღებთ საწყის უწყვეტ სიგნალს:

$$\begin{aligned}
 X(t) &= \int_{-\infty}^{\infty} X(\tau) h(t-\tau) d\tau = \int_{-\infty}^{\infty} \sum_{i=-\infty}^{\infty} X(i\Delta t) \delta(t-i\Delta t) \text{sinc}(2\pi F_m(t-\tau)) d\tau \\
 &= \sum_{i=-\infty}^{\infty} X(i\Delta t) \int_{-\infty}^{\infty} \delta(t-i\Delta t) \text{sinc}(2\pi F_m(t-\tau)) d\tau = \sum_{i=-\infty}^{\infty} X(i\Delta t) \text{sinc}(2\pi F_m(t-i\Delta t)). \quad (1.17)
 \end{aligned}$$

უწყვეტი $X(t)$ ფუნქციის დისკრეტიზაციისა და ანათვლების $X(t)$ დისკრეტული მიმდევრობიდან მისი აღდგენის პროცესი წარმოდგენილია 1.4 ნახაზზე.

ამრიგად, ფუნქციის ანათვლების $X(i\Delta t)$ დისკრეტული მიმდევრობის საშუალებით ყოველთვის შესაძლებელია საწყისი უწყვეტი $X(t)$ ფუნქციის აღდგენა, თუ ანათვლები აღებული იყო $\Delta t \leq 1/2F_m$ ინტერვალით (პერიოდით). ეს კი იმაზე მეტყველებს, რომ უწყვეტ და დისკრეტულ სიგნალებს შორის არ არსებობს პრინციპიული განსხვავება. შეზღუდული სპექტრის მქონე ნებისმიერი უწყვეტი სიგნალი (ხოლო ყველა რეალურ სიგნალს გააჩნია შეზღუდული სპექტრი) შეიძლება გარდაიქმნას დისკრეტულ მიმდევრობად, ხოლო შემდეგ თავისი დისკრეტული ანათვლების საშუალებით აღდგენილ იქნას აბსოლუტური სიზუსტით. ეს გარემოება იძლევა: უწყვეტი შეტყობინებების წყაროების დისკრეტული მიმდევრობების წყაროების სახით განხილვის საშუალებას; იქ, სადაც აუცილებელი და მოსახერხებელია, შეიძლება გადავიდეთ დისკრეტული შეტყობინებების ანალიზზე; უწყვეტი შეტყობინებების გადაცემა შეიძლება განხორციელდეს დისკრეტული ფორმით და ასე შემდეგ.



ნახ. 14. უწყვეტი $X(t)$ ფუნქციის დროითი დისკრეტიზაციის პროცესის ამსახველი დროითი და სპექტრული დიაგრამები

1.3.1. რეალური სიბნალების დისკრეტიზაციის პრაქტიკული საკითხები

კავშირის არხებით გადასაცემი შეტყობინებები (სიტყვა, მუსიკა, სატელევიზიო სიგნალი, ტელემეტრული მონაცემები და ა.შ.) პრაქტიკულად შეზღუდული სპექტრის მქონე ფუნქციებია. მაგალითად, ზედა F_m სიხშირე დაახლოებით ტოლია: მეტყველებისათვის – 3,5 კჰც, მუსიკისათვის – 10...12 კჰც (დამაკმაყოფილებელი აღწარმოების შემთხვევაში), სატელევიზიო სიგნალებისათვის – 6 მჰც.

გარკვეული არაკორექტულობა მდგომარეობს იმაში, რომ ანათვლების თეორემა დამტკიცებულია დროის შეუზღუდავი $t \in (-\infty, \infty)$ ინტერვალისათვის. შესაბამისად $\{X(i \Delta t), i=0, \pm 1, \pm 2, \dots\}$ ანათვლები ქმნიან უსასრულო მიმდევრობას. მაგრამ რეალურ პირობებში $X(t)$ შეტყობინებებს აქვთ დასაწყისი და დასასრული და, შესაბამისად, სასრულო $T < \infty$ ხანგრძლივობა. სპექტრის ფინიტურობისა და შეტყობინების სასრულო ხანგრძლივობის პირობები, მკაცრად რომ ვთქვათ, ერთმანეთის შეუსაბამოა. თეორიულად სასრულო ხანგრძლივობის ფუნქციის სპექტრს $F \in (-\infty, \infty)$ დიაპაზონის სიხშირის ნებისმიერი მნიშვნელობისათვის გააჩნია ნულისაგან განსხვავებული მნიშვნელობები. მაშინ დისკრეტიზაციის Δt ბიჯის (პერიოდის) ნებისმიერი შერჩევისას სპექტრის გვერდითი ზოლები (იხ. ნახ. 1.4) გადაიფარებიან (გადაიკვეთებიან), რის გამოც დაბალი სიხშირეების თუნდაც იდეალური ფილტრის გამოსასვლელზე, რომლის ჩამოჭრის სიხშირეა $F=1/2\Delta t$, $X^*(t)$ სიგნალი აღდგება საწყისი $X(t)$ სიგნალისაგან განსხვავებულად. ჯერ ერთი, წაიკვეთება სპექტრის ის სიხშირული შემდგენები, რომლებისთვისაც $f > F$, და მეორე, ფილტრის გატარების ზოლში ექცევა სპექტრის პერიოდული გაგრძელების "კუდები".

ამასთანავე, ყოველთვის შესაძლებელია დისკრეტიზაციის Δt ბიჯის (ანუ სიგნალის სპექტრის ზედა ზღვრული $F_m=1/2\Delta t$ სიხშირის) ისეთნაირი შერჩევა, რომლის დროსაც სპექტრის $f > 1/2\Delta t$ სიხშირეებზე არსებულ წაკვეთად "კუდებში" თავმოყრილი E_d ენერგია მთელი სიგნალის E_x ენერგიასთან შედარებით იმდენად მცირე აღმოჩნდეს, რომ შესაძლებელი იყოს მისი უგულვებელყოფა. ფილტრის გამოსასვლელზე აღდგენილი $X^*(t)$ სიგნალის შეცდომა დამოკიდებულია E_d/E_x ფარდობაზე და იგი Δt -ს (ან $F=1/2\Delta t$ -ს) სათანადოდ შერჩევის შემთხვევაში შეიძლება გაეხად თ ნებისმიერ მოცემულ მნიშვნელობაზე ნაკლები. სრულიად ნათელია, რომ თუ შეტყობინებათა დროითი დისკრეტიზაციით გამოწვეული

დამახინჯებები აღმოჩნდება გაცილებით ნაკლები იმ დამახინჯებებთან შედარებით, რომლებიც გამოწვეულია კავშირის არხში არსებული და ინფორმაციის გადაცემის მოცემული სისტემის ტექნიკური პირობებით განსაზღვრული ხელშეშლების დასაშვები სიდიდით, მაშინ ასეთი დამახინჯებები არაარსებითია და ამიტომ შესაძლებელია მათი უგულვებელყოფა.

ამრიგად, მიახლოებით შეიძლება მივიღოთ, რომ რეალური შეტყობინებები სასრულო T ხანგრძლივობისაა და იმავდროულად მათი სპექტრები შეზღუდულია F_m სიხშირით. ამასთანავე (1.17) უსასრულო მწკრივი გარდაიქმნება იმ n რაოდენობის არანულოვანი ანათვლების სასრულო მწკრივად, რომელიც დაახლოებით შეტყობინების ხანგრძლივობის დისკრეტიზაციის ინტერვალთან (პერიოდთან) ფარდობის ტოლია:

$$n \approx T/\Delta t = 2F_m T. \quad (1.18)$$

დისკრეტიზაციის თეორემის ძირითადი ფორმულები იმ სიგნალებისათვის, რომლებიც ნულისგანაა განსხვავებული სასრულო $t \in (0, T)$ ინტერვალში, შეიძლება წარმოვადგინოთ შემდეგი სახით:

$$X(t) = \sum_{i=0}^{n-1} X(i\Delta t) \text{sinc}(2\pi F_m(t - i\Delta t)), \quad n = 2F_m T; \quad (1.19)$$

$$E_x = \int_0^T X^2(t) dt = (1/2F_m) \sum_{i=0}^{n-1} X^2(i\Delta t); \quad (1.20)$$

$$E_x \approx 2 \int_{1/2\Delta t}^{\infty} |F_x(f)|^2 df. \quad (1.21)$$

და ბოლოს, იმ შემთხვევაში, როდესაც სიგნალი $\{X(t), t \in (0, T)\}$ მოცემულია სასრულო რაოდენობის $X(0), X(\Delta t), \dots, X(k\Delta t)$ ანათვლების სახით, მაშინ (1.17) – (1.19) ფორმულებში, შესაბამისი ზუსტი ფორმულებისაგან განსხვავებით, მიზანშეწონილი იქნებოდა მიახლოებითი ტოლობის ნიშნის (\approx) დასმა. თუმცა, ჩვეულებრივ, ამას არ აკეთებენ.

კიდევ ერთ მიახლოებას, რომელიც სინამდვილეში ვერ შესრულდება, წარმოადგენს აღმდგენი ფილტრის $H(f)$ ამპლიტუდურ-სიხშირული მახასიათებლის "იდეალურობა". საქმე იმაშია, რომ იდეალურად სწორკუთხა ამპლიტუდურ-სიხშირული მახასიათებლის მქონე ფილტრს გააჩნია უსასრულო ხანგრძლივობის იმპულსური გარდამავალი მახასიათებელი და მისი პრაქტიკული რეალიზაცია შეუძლებელია. სასრული იმპულსური გარდამავალი მახასიათებლის მქონე ფილტრებს გააჩნიათ *თეორიულად* უსასრულო ზოლი. რთული არაა იმის

ჩვენება, რომ აღმდგენი ფილტრის იმპულსური გარდამავალი მახასიათებლის სასრულო ხანგრძლივობის $X^*(t)$ სიგნალზე გავლენას იგივე ხასიათი აქვს, როგორც $X(t)$ ფუნქციის დაკვირვების ინტერვალის შეზღუდულობას.

ამრიგად, მოცემული ამპლიტუდურ-სიხშირული მახასიათებლის მქონე დაბალი სიხშირეების ფილტრისათვის ყოველთვის შეიძლება შეირჩეს დისკრეტიზაციის ისეთი Δt ბიჯი (პერიოდი), რომ მისი ამპლიტუდურ-სიხშირული მახასიათებლის $f > 1/2\Delta t$ სიხშირეებზე, ანუ ამ მახასიათებლის "კუდებში" გაჟონილი E_A ენერგია იმდენად მცირე იყოს მთლიანი სიგნალის E_x ენერგიასთან შედარებით, რომ შესაძლებელი იყოს მისი უგულვებელყოფა. ამის გამო პრაქტიკაში რეალური $X(t)$ შეტყობინებების დისკრეტიზაციის ბიჯს (პერიოდს) იღებენ რამდენადმე ნაკლებს, ხოლო შესაბამისად, დისკრეტიზაციის სიხშირეს – რამდენადმე მეტს (უკიდურეს შემთხვევაში 30–50%-ით) თეორიულ ვარიანტთან შედარებით.

14. ორგანზომილებიანი სიბნალების (გამოსახულებათა) დისკრეტიზაცია

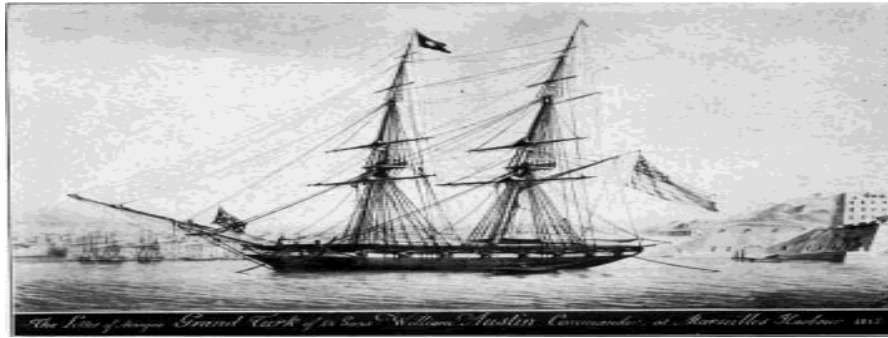
ინფორმაციის გადაცემის რადიოტექნიკური სისტემების საშუალებით გადასაცემ შეტყობინებათა სულ უფრო დიდ ნაწილს იკავებს სიგნალები, რომლებიც არა მხოლოდ დროის $\lambda(t)$ (სიტყვა, მუსიკა და სხვა), არამედ მთელი რიგი სხვა ცვლადების ფუნქციაა (მაგალითად, $\lambda(x,y)$, $\lambda(x,y,t)$ – სტატიკური და დინამიური გამოსახულებები, ფიზიკური ველების რუკები და ა.შ.). ამასთან დაკავშირებით ბუნებრივია კითხვა – შესაძლებელია თუ არა ასეთი მრავალგანზომილებიანი სიგნალების (რამდენიმე ცვლადის ფუნქციების) დისკრეტიზაცია ისე, როგორც ეს კეთდება დროითი სიგნალებისათვის (ან ერთი ცვლადის სხვა ფუნქციებისათვის)?

აღნიშნულ კითხვაზე პასუხს იძლევა ორგანზომილებიანი (ან ზოგადად მრავალგანზომილებიანი) სიგნალების შესახებ დისკრეტიზაციის თეორემა, რომლის მიხედვითაც: *ორი ცვლადის $\lambda(x,y)$ ფუნქცია, რომლისთვისაც ფურიეს ორგანზომილებიანი გარდასახვა*

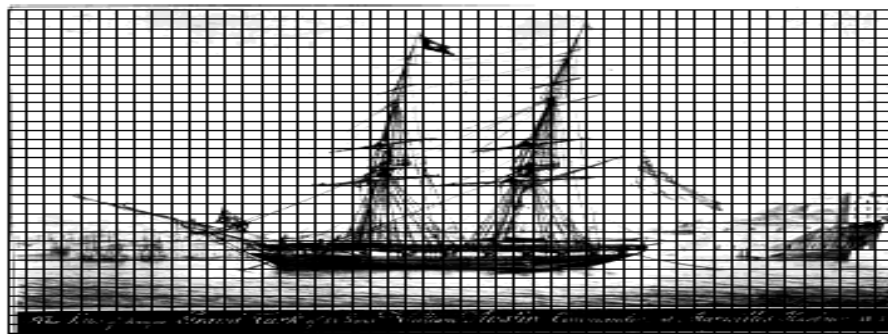
$$FF\{\lambda(x,y)\} = \iint_{-\infty}^{\infty} \lambda(x,y) \exp(-j2\pi f_x x) \exp(-j2\pi f_y y) dx dy \quad (1.22)$$

ნულის ტოლია $f_x \geq f_{x,max}$ და $f_y \geq f_{y,max}$ პირობების შესრულებისას, ცალსახად განისაზღვრება x და y ცვლადების სიბრტყის თანაბრად დაშორებულ

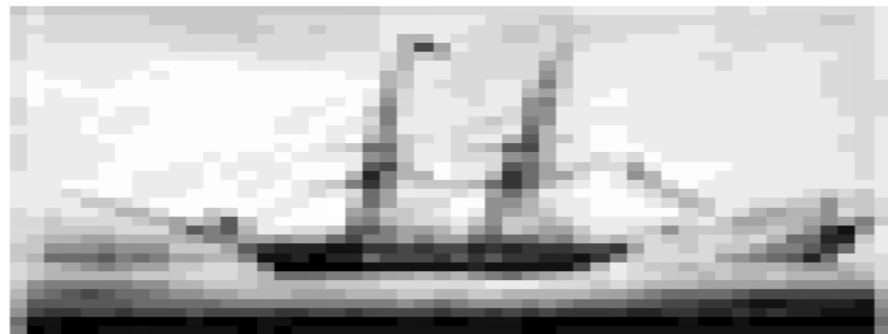
წერტილებში ამ ფუნქციის მნიშვნელობებით, თუ დისკრეტიზაციის ინტერვალი აკმაყოფილებს $\Delta x \leq 1/2f_x$, $\Delta y \leq 1/2f_y$ პირობას. ორგანზომილებიანი ფუნქციის დისკრეტიზაციის პროცედურა ილუსტრირებულია 1.5 ა, ბ, გ ნახაზებზე.



ა



ბ



გ

ნახ. 1.5. ორგანზომილებიანი გამოსახულების დროითი დისკრეტიზაციის ნიმუში

ორგანზომილებიანი დისკრეტიზაციის თეორემის დამტკიცება, ისევე როგორც ერთგანზომილებიანის შემთხვევაში, ემყარება ცალსახა შესაბამისობას სიგნალებსა და მათ სპექტრებს შორის: ერთნაირ გამოსახულებებს (ორგანზომილებიან ფუნქციებს) შეესაბამება ერთნაირი სპექტრები და, პირიქით, თუ ორი ფუნქციის სპექტრები ერთნაირია, მაშინ ეს ფუნქციებიც ერთნაირია (ისინი ერთმანეთის ტოლია).

ორგანზომილებიანი დისკრეტიზირებული ფუნქციის ფურიეს $FF\{i\Delta x, j\Delta y\}$ გარდასახვა (სპექტრი) $FF\{i\Delta x, j\Delta y\}$ მიიღება საწყისი უწყვეტი $\lambda(x,y)$ ფუნქციის სპექტრის პერიოდული გაგრძელებით სიხშირული სიბრტყის $(k\Delta f_x, l\Delta f_y)$ წერტილებში (ნახ. 1.6), სადაც f_x და f_y ეგრეთ წოდებული "სივრცითი სიხშირეებია", რომლებიც ჩვეულებრივი "დროითი" სიხშირის ანალოგებია და ასახავენ ორგანზომილებიანი $\lambda(x,y)$ ფუნქციის ცვლილებების სიჩქარეს შესაბამისი $(x$ და $y)$ კოორდინატების მიხედვით (მსხვილ ფრაგმენტებს შეესაბამება დაბალი სიხშირეები, ხოლო წვრილ დეტალებს – მაღალი).

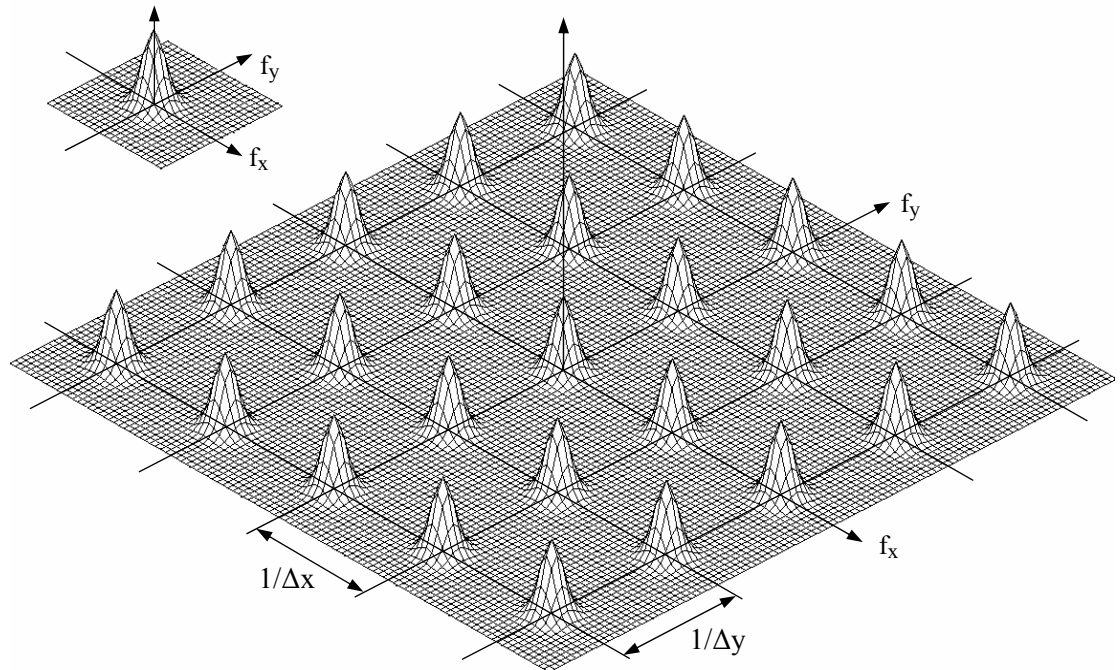
ყოველივე ეს ანალიზურად შეიძლება ჩაიწეროს შემდეგი სახით:

$$FF\{i\Delta x, j\Delta y\} = (1/\Delta_x)(1/\Delta_y) \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \lambda(f_x - k\Delta f_x, f_y - l\Delta f_y). \quad (1.23)$$

1.6 ნახაზიდან ჩანს, რომ თუ სრულდება $FF\{i\Delta x, j\Delta y\}$ სპექტრის პერიოდული გაგრძელებების გადაუკვეთადობის პირობა, რაც სამართლიანია $\Delta x \leq 1/2 f_x \max$ და $\Delta y \leq 1/2 f_y \max$ პირობების შესრულებისას, მაშინ იდეალური ორგანზომილებიანი

$$H(f_x, f_y) = \begin{cases} \Pi(f_x / \Delta f_x) \cdot \Pi(f_y / \Delta f_y), & |f_x| \leq 1/2 \Delta x, |f_y| \leq 1/2 \Delta y, \\ 0, & |f_x| > 1/2 \Delta x, |f_y| > 1/2 \Delta y. \end{cases} \quad (1.24)$$

სახის სიხშირული მახასიათებლის მქონე დაბალი სიხშირეების ფილტრის საშუალებით დისკრეტიზირებული $FF\{i\Delta x, j\Delta y\}$ ფუნქციის საშუალებით აბსოლუტურად ზუსტადაა შესაძლებელი საწყისი უწყვეტი $FF\{\lambda(x,y)\}$ და, შესაბამისად, თვით ფუნქციის აღდგენა.



ნახ. 1.6. დისკრეტიზირებული ორგანზომილებიანი ფუნქციის ფურიეს გარდაქმნის შედეგი

ამრიგად, ზემოაღნიშნული მსჯელობიდან ჩანს, რომ არ არსებობს პრინციპიალური განსხვავება ერთგანზომილებიანი და ორგანზომილებიანი (მრავალგანზომილებიანი) ფუნქციების დისკრეტიზაციებს შორის. ორივე შემთხვევაში დისკრეტიზაციის შედეგს წარმოადგენს ფუნქციის ანათვლების ერთობლიობა. განსხვავება შეიძლება იყოს მხოლოდ დისკრეტიზაციის ბიჯის სიდიდეში, ანათვლების რაოდენობასა და მათი მიმდევრობის რიგითობაში.

1.5. შეტყობინებათა დაკვანტვა. დაკვანტვის შეცდომა

წინა პარაგრაფებში დასაბუთებულია, რომ პრაქტიკულად ნებისმიერი $\lambda(t)$ ($\lambda(x,y)$) შეტყობინების გადაცემა შეიძლება შევცვალოთ მისი იმ $\lambda_i = \lambda(i\Delta t)$ ანათვლების ანუ რიცხვების გადაცემით, რომლებიც ერთმანეთს მიყვებიან დისკრეტულობის $\Delta t \leq 1/2F_m$ ($\Delta x \leq 1/2f_x$, $\Delta y \leq 1/2f_y$) ინტერვალით. შედეგად ამისა $\lambda(t)$ შეტყობინების შესაძლო მნიშვნელობათა უწყვეტი (უსასრულო) სიმრავლეს ენაცვლება მისი დისკრეტული $\{\lambda(i\Delta t)\}$ მნიშვნელობების სასრულო რაოდენობა. თუმცა თვით ამ რიცხვებს გააჩნია დონეთა (მნიშვნელობათა) უწყვეტი შკალა, რის გამოც მათი აბსოლუტურად ზუსტი წარმოდგენისათვის, მაგალითად,

ათობით (ან ორობით) ფორმაში თეორიულად თანრიგების უსასრულო რაოდენობაა საჭირო. ამასთანავე, პრაქტიკაში სულაც არაა აუცილებელი ნებისმიერი რიცხვის (მათ შორის λ -ს) მნიშვნელობათა წარმოდგენა აბსოლუტური სიზუსტით.

ჯერ ერთი, თავად შეტყობინებათა წყაროებს გააჩნიათ შეზღუდული დინამიური დიაპაზონი და ისინი საწყის შეტყობინებებს აფორმირებენ გარკვეული სიდიდის დამახინჯებებითა და შეცდომებით. ისინი შეიძლება იყოს დიდი ან მცირე, მაგრამ შეტყობინების აღწარმოების აბსოლუტური სიზუსტის მიღწევა მაინც შეუძლებელია.

მეორეც, კავშირის არსებით შეტყობინებათა გადაცემა ყოველთვის ხდება სხვადასხვა სახის ხელშეშლათა ფონზე. ამიტომ მიღებული შეტყობინება (შეტყობინების $\lambda^*(t)$ ან A^* შეფასება) გარკვეული ხარისხით ყოველთვის განსხვავებულია გადაცემული შეტყობინებისაგან, რაც იმას ნიშნავს, რომ კავშირის არსში ხელშეშლების არსებობისას პრაქტიკულად შეუძლებელია შეტყობინებათა აბსოლუტურად ზუსტი გადაცემა.

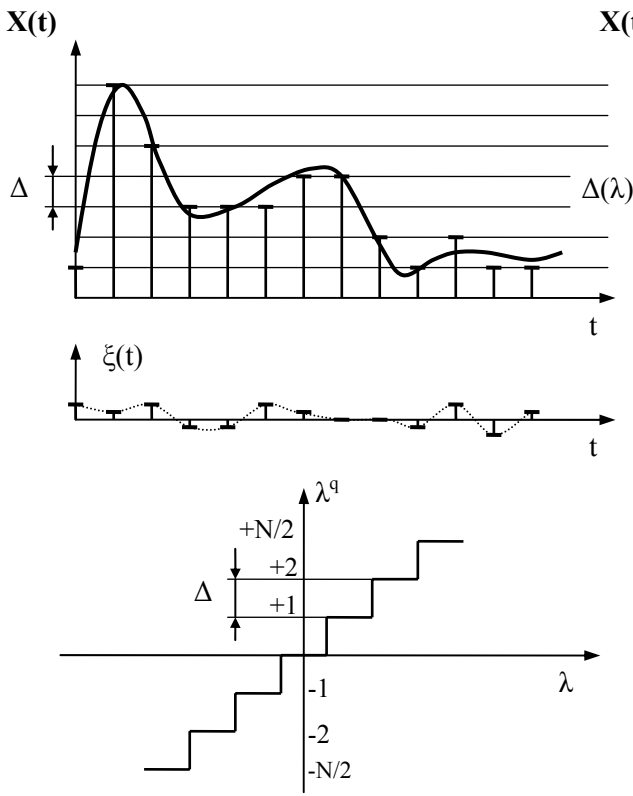
და ბოლოს, შეტყობინებათა გადაცემა ხდება მიმღების მიერ მათი აღქმისა და გამოყენების მიზნით. ინფორმაციის მიმღებს კი წარმოადგენენ ადამიანის შეგრძნების ორგანოები, ამა თუ იმ ოპერაციის შემსრულებელი მექანიზმები და სხვა., რომლებსაც სასრულო გარჩევის უნარიანობა გააჩნიათ, ანუ ისინი ვერ შენიშნავენ უმნიშვნელო განსხვავებას საწყისი შეტყობინების აბსოლუტურად ზუსტ და აღწარმოებული (აღდგენილი) შეტყობინების მიახლოებულ მნიშვნელობებს შორის. მგრძნობიარობის ზღვარი სხვადასხვა შემთხვევაში შეიძლება იყოს სხვადასხვა, თუმცა იგი ყოველთვის არსებობს.

აღნიშნული შენიშვნების გათვალისწინებით შეტყობინებათა დისკრეტიზაციის პროცესი შეიძლება გავაგრძელოთ, კერძოდ კი მოვახდინოთ მისი λ_i ანათვლების დაკვანტვა (დისკრეტიზაცია მნიშვნელობათა მიხედვით).

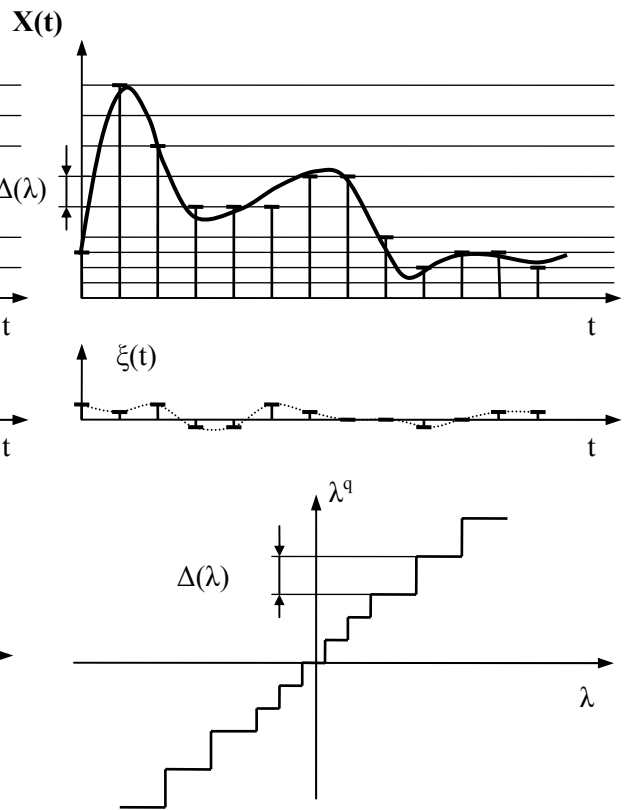
დაკვანტვის შედეგად $\lambda_i \in (\lambda_{min}, \lambda_{max})$ ანათვლების მნიშვნელობათა უწყვეტი სიმრავლე იცვლება $A\{\lambda_i\}$ აღფაბეტის დისკრეტული $\{\lambda_{(1)}, \dots, \lambda_{(m)}\}$ სიმრავლით. ამის შედეგად λ_i რიცხვების მნიშვნელობები შეიზღუდება მათი მიახლოებული (უახლოეს დასაშვებ დონემდე დამრგვალებული) მნიშვნელობებით. მეზობელ დასაშვებ λ_i , ანუ დაკვანტვის, დონეებს შორის $\Delta = \lambda_{i+1} - \lambda_i$ ინტერვალს დაკვანტვის ბიჯს უწოდებენ.

ცნობილია თანაბარი და არათანაბარი დაკვანტვა. უმრავლეს შემთხვევაში გამოიყენება თანაბარი დაკვანტვა (ნახ. 1.7), რომელიც ჩვენს მიერ დაწვრილებით

იქნება განხილული. ამ შემთხვევაში დაკვანტვის ბიჯი $\Delta = \lambda_i - \lambda_{i-1} = \text{Const}$. თუმცა ზოგჯერ გარკვეული უპირატესობებს იძლევა არათანაბარი დაკვანტვა (ნახ. 18), რომლის დროსაც დაკვანტვის Δ_i სხვადასხვაა სხვადასხვა λ_i -სათვის.



ნახ. 17. თანაბარი დაკვანტვის ილუსტრაცია



ნახ. 18. არათანაბარი დაკვანტვის ილუსტრაცია

დაკვანტვა იწვევს შეტყობინებათა დამახინჯებას. თუ $\lambda_i = \lambda(i\Delta t)$ ანათელის დაკვანტვით მიღებული დაკვანტული შეტყობინებაა λ_{iq} , მაშინ

$$\lambda_{iq} = \lambda_i + \xi_i, \quad (1.25)$$

სადაც ξ_i სხვაობაა დაკვანტულ λ_{iq} შეტყობინებასა (უახლოეს დასაშვებ მნიშვნელობასა) და ელემენტალური λ_i შეტყობინების ნამდვილ მნიშვნელობას შორის და მას დაკვანტვის ხმაურს უწოდებენ. ინფორმაციის გადაცემის პროცესზე დაკვანტვის ხმაური არსებითად იგივე გავლენას ახდენს, რასაც ხელშეშლები კავშირის არხში. ხელშეშლები, ისევე როგორც დაკვანტვა, იწვევს კავშირის სისტემის მიმღებ ნაწილში მიღებული λ^* შეფასებების გარკვეული სიდიდით განსხვავებას ნამდვილ λ_i მნიშვნელობასთან შედარებით.

იმის გამო, რომ შეტყობინებათა დაკვანტვა იწვევს შეცდომების წარმოქმნასა და ინფორმაციის გარკვეული ნაწილის დაკარგვას, შეიძლება

განისაზღვროს ასეთი დანაკარგების ფასი $d(\lambda, \lambda_q)$ და დაკვანტვით გამოწვეული შეცდომის საშუალო სიდიდე $D(q)$:

$$D(q) = \langle d(\lambda, \lambda_q) \rangle. \quad (1.26)$$

ხშირად დანაკარგების ფუნქციის (დანაკარგების ფასის) სახით იყენებენ შემდეგი სახის კვადრატულ ფუნქციას:

$$d(\lambda, \lambda_q) = |\lambda - \lambda_q|^2. \quad (1.27)$$

ამ შემთხვევაში დაკვანტვის შეცდომების საზომად გამოიყენება ამ შეცდომების დისპერსია. Δ ბიჯის მქონე თანაბარი N -დონიანი დაკვანტვის შემთხვევაში დაკვანტვის შეცდომების დისპერსია განისაზღვრება შემდეგნაირად:

$$D(q) \cong \sum_{i=1}^N \int_{\lambda_{iq} - \Delta/2}^{\lambda_{iq} + \Delta/2} (\lambda - \lambda_q)^2 f(\lambda) d(\lambda) = (1/N) \sum_{i=1}^N \int_{-\Delta/2}^{\Delta/2} \xi^2 f(\xi) d\xi. \quad (1.28)$$

დაკვანტვის შეცდომის აბსოლუტური მნიშვნელობა არ აღემატება დაკვანტვის ბიჯის ნახევარს $\Delta/2$ და მაშინ დაკვანტვის დონეების საკმარისად დიდი N რაოდენობისა და Δ -ს მცირე სიდიდის შემთხვევაში დაკვანტვის შეცდომების ალბათობათა განაწილების $f(\xi)$ სიმკვრივე შეიძლება ჩაითვალოს თანაბრად $+ \Delta/2 \dots - \Delta/2$ ინტერვალზე:

$$f(\xi) = \begin{cases} 1/\Delta, & |\xi| \leq \Delta/2, \\ 0, & |\xi| \geq \Delta/2. \end{cases} \quad (1.29)$$

ამის შედეგად დაკვანტვის შეცდომის აბსოლუტური $D(q) = \sigma_q^2$ მნიშვნელობა განისაზღვრება თანაფარდობით

$$D(q) = \sigma_q^2 = (1/N\Delta) \sum_{i=1}^N \int_{-\Delta/2}^{\Delta/2} \xi^2 d\xi = N(1/N\Delta) \Delta^3 / 12 = \Delta^2 / 12 \quad (1.30)$$

და დაკვანტვის Δ ბიჯის შესაბამისი შერჩევით იგი შეიძლება გავხადოთ რაც შეიძლება მცირე ან წინასწარ მოცემული (დასაშვები) სიდიდის ტოლი.

შეტყობინებათა ანათვლების გადაცემის სიზუსტის შესახებ შეიძლება გამოითქვას იგივე მოსაზრებები, რაც დროითი დისკრეტიზაციის შეცდომათა შესახებ: დაკვანტვის ხმაურები ან დაკვანტვით გამოწვეული დამახინჯებები არაა მნიშვნელოვანი, თუ ეს დამახინჯებები ნაკლებია იმ შეცდომებთან შედარებით, რომლებიც განპირობებულია ხელშეშლებითა და დასაშვები ტექნიკური პირობებით. მაგალითად, სიტყვისა და მუსიკის გადაცემისას, დამახინჯებები პრაქტიკულად შეუმჩნეველია, თუ ყველა ანათვალს შემთხვევითად შევცვლით 0,1...1%-ით, გამოსახულებათა გადაცემისას – 1%-ით და ა.შ. პროფესიონალ

ექსპერტსაც კი არ შეუძლია შეამჩნიოს დამახინჯებები მუსიკალურ ნაწარმოებში, თუ დაკვანტვა ხორციელდება 0,001%-ზე უკეთ (დაკვანტვის დონეთა რაოდენობა $N > 100000$, ანათვლების წარმოდგენის სიზუსტეა 16...17 ორობითი თანრიგი). ტელემეტრულ სისტემებში დაკვანტვის დონეთა რაოდენობა დამოკიდებულია ინფორმაციის აღწარმოების სიზუსტის საჭირო სიდიდეზე და, აგრეთვე, იმ მიმწოდების სიზუსტეზე, რომლებიც აგროვებენ ამ ინფორმაციას. ამასთან დაკვანტვისას მიმწოდების მიერ მიღწევადი ან მოთხოვნილი სიზუსტისადმი გადამეტება არაა მიზანშეწონილი აპარატურის სირთულისა და გადაცემისათვის დანახარჯების გაზრდის გამო. უფრო მეტიც, კავშირის ხელშეშლებიანი არხით გადაცემისას შეიძლება წარმოიქმნას ისეთი სიტუაციები, როდესაც შეტყობინების λ^* ; შეფასების აღწარმოების (აღდგენის) ხარისხი გადამცემში ამ შეტყობინების უფრო უხეშად დაკვანტვისას აღმოჩნდეს მნიშვნელოვნად უკეთესი, ვიდრე ზუსტი დაკვანტვის შემთხვევაში. ამ საკმაოდ არანათელ, მაგრამ ინფორმაციის გადაცემის ზოგადი თეორიიდან გამომდინარე მოვლენაზე მომავალში შევჩერდებით უფრო დაწვრილებით.

ამრიგად, ნახვენებია, რომ პრაქტიკულად ნებისმიერი $\lambda(t)$ ($\{ \lambda(x,y) \}$) შეტყობინების წინასწარ მოცემული ნებისმიერი სიზუსტით გადაცემა შეიძლება შეიცვალოს დისკრეტულობის $\Delta t \leq 1/2F_m$ ($\Delta x \leq 1/2f_{x,max}$, $\Delta y \leq 1/2f_{y,max}$) ინტერვალით დაცილებული მთელი $\lambda_{iq} = \lambda_q(i \Delta t)$ რიცხვების გადაცემით. ამით $\lambda(t)$ ($\{ \lambda(x,y) \}$) შეტყობინების შესაძლო მნიშვნელობათა უწყვეტი (უხასრულო) სიმრავლე შეიცვლება $A\{ \lambda_{iq} \}$ ($i=1,2,...,N$) ალფაბეტის მთელი რიცხვების ხასრულო სიმრავლით.

სხვა სიტყვებით რომ ვთქვათ, ახლა შეიძლება სიგნალებთან ისე ვიმუშაოთ, როგორც ციფრებთან. ეს კი იძლევა მათი დამუშავებისა და ანალიზისათვის ანალოგურ ფორმაში პრაქტიკულად არარეალიზებადი ნებისმიერი სირთულის ციფრული ალგორითმების, ინფორმაციის გადაცემის სისტემებში ციფრული მეთოდებისა და თანამედროვე ციფრული ინტეგრალური ტექნოლოგიების გამოყენების საშუალებას.

ამრიგად, ჩვენ გავარკვეით, რომ რადიოელექტრონულ სისტემებში ინფორმაციის მატარებლებს ან გადამტანებს წარმოადგენს ელექტრული სიგნალები, რომლებსაც აფორმირებენ ამ ინფორმაციის წყაროები. იმ შემთხვევებშიც კი, როდესაც პირველადი ინფორმაცია არაელექტრული ბუნებისაა (სიტყვა, მუსიკა, გამოსახულებები, ტექსტები, მონაცემთა პაკეტები და ა.შ.), იგი საბოლოოდ მაინც გარდაიქმნება ელექტრულ სიგნალებად და შემდეგ

შეინახება ან გადაიცემა კავშირის არხებით. ეს სიგნალები ჩვეულებრივ უწყვეტი ხასიათისაა, ანუ ისინი განსაზღვრულია დროის ნებისმიერი მომენტისათვის ან თავისი არსებობის უსასრულო რაოდენობის წერტილებში. გაცილებით მოსახერხებელია სასრულო ზომის მქონე მონაცემებთან, მაგალითად, რიცხვების სასრულო ზომისა და შეზღუდული რაოდენობის თანრიგების მქონე მასივებთან "ურთიერთობა". სწორედ დისკრეტიზაციის ზემოთ განხილული თეორემა იძლევა ამის შესაძლებლობას.

1.6. ინფორმაციის რაოდენობა. შეტყობინებათა წყაროს ენტროპია

შეტყობინებათა სხვადასხვა წყაროების ერთმანეთთან შედარებისათვის საჭიროა შემოტანილი იყოს რაღაც რაოდენობრივი საზომი, რომელიც შესაძლებელს გახდიდა ობიექტურად შეფასებულიყო შეტყობინებაში არსებული ინფორმაცია. ასეთი საზომი პირველად იქნა შემოტანილი კლოდ შენონის მიერ 1948 წელს, ხოლო შემდეგში იგი უფრო მკაცრად იქნა განსაზღვრული ა.ი.ხინჩინის მიერ. განვიხილოთ შენონის ინფორმაციული მიდგომის საფუძვლები.

ყოველგვარ ინფორმაციას მომხმარებელი ღებულობს შეტყობინების მიღების შემდეგ, ანუ გამოცდილების შედეგად. მიმღებ მხარეს მიღებულ შეტყობინებას სასარგებლო ინფორმაცია გააჩნია მხოლოდ იმ შემთხვევაში, თუ არსებობს გაურკვევლობა წყაროს მდგომარეობის შესახებ. თუ ცდა დამთავრდება მხოლოდ ერთი შედეგით და დამკვირვებელმა წინასწარ იცის ცდის შედეგი, მაშინ მაშინ ის ვერავითარ ინფორმაციას ვერ ღებულობს. მაგალითად, თუ შეგატყობინებენ, რომ მზე აღმოსავლეთიდან ამოდის, მაშინ ეს შეტყობინება არავითარ ინფორმაციას არ შეიცავს, ვინაიდან ყველამ იცის, რომ ეს ნამდვილად ასეა. ისეთ მოვლენაში, როგორცაა მზის ყოველდღიური ამოსვლა აღმოსავლეთიდან, არაფერია გაურკვეველი. ასეთი მოვლენის აღბათობა ერთის ტოლია და შეტყობინების მიერ მოტანილი ინფორმაციის რაოდენობა ნულის ტოლია. ინფორმაცია გაჩნდება მხოლოდ მაშინ, როდესაც წყაროს ექნება შესაძლო მდგომარეობათა უკიდურეს შემთხვევაში ერთზე მეტი რაოდენობა.

განვიხილოთ წყარო, რომელიც გასცემს ურთიერთდამოუკიდებელი დისკრეტული შეტყობინებების $\{\lambda_i\}$ მიმდევრობას, რომელთაგან თითოეულ მათგანს შეტყობინებათა $A(\lambda_i) = \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_K$ ალფაბეტიდან (K —წყაროს ალფაბეტის ზომაა) შეარჩევენ შემთხვევითი კანონით. ასეთი წყაროს მიერ გამომუშავებულ შეტყობინებებს ვუწოდოთ მარტივი შეტყობინებები.

ყოველი λ_i ელემენტარული შეტყობინება ამ შეტყობინების მიმღებისათვის შეიცავს გარკვეულ ინფორმაციას. განვსაზღვროთ ამ ინფორმაციის რაოდენობრივი ზომა და გავარკვიოთ, თუ რაზეა ის დამოკიდებული.

მანამ, სანამ დამყარდება კავშირი, ინფორმაციის მიმღებს ყოველთვის გააჩნია მეტი ან ნაკლები გაურკვეველობა იმიასთან დაკავშირებით, თუ შეტყობინებათა შესაძლო რაოდენობიდან რომელი λ_i შეტყობინება იქნება გადაცემული.

სრულიად ცხადია, რომ ამ გაურკვეველობის (განუსაზღვრელობის), ანუ λ_i შეტყობინების მოულოდნელობის ხარისხი, დამოკიდებულია ამა თუ იმ შეტყობინების გადაცემის ალბათობაზე. მაგალითად, თუ რომელიმე λ_i შეტყობინების გადაცემის ალბათობა მაღალია, მაშინ გადაცემამდე მომხმარებელმა თითქმის იცის ის, თუ რომელი შეტყობინება იქნება გადაცემული, და მისი მიღება მომხმარებელს თითქმის არ მოუტანს რაიმე ახალ ინფორმაციას.

ამრიგად, ცხადია, რომ ელემენტარულ λ_i შეტყობინებაში არსებული ინფორმაციის რაოდენობა წარმოადგენს ამ შეტყობინების გადაცემის $P(\lambda_i)$ ალბათობის ფუნქციას:

$$J(\lambda_i) = \varphi\{P(\lambda_i)\}, \quad (1.31)$$

განვსაზღვროთ ამ φ ფუნქციის სახე. ამისათვის მოვითხოვოთ, რომ ინფორმაციის რაოდენობის $J(\lambda_i)$ ზომამ დააკმაყოფილოს შემდეგი ორი ინტუიტიური თვისება (მოთხოვნა):

1. თუ λ_i შეტყობინების ამორჩევა წინასწარაა განსაზღვრული ($P(\lambda_i)=1$ – გაურკვეველობა არ არსებობს), მაშინ ამ შეტყობინებაში არსებული ინფორმაციის რაოდენობა ნულის ტოლია: $J(\lambda_i)=\varphi\{1\}=0$.

2. თუ წყარო მიმღევრობით ირჩევს λ_i და λ_j შეტყობინებებს და ასეთი ამორჩევის ალბათობა $P(\lambda_i, \lambda_j)$ არის λ_i და λ_j შეტყობინებების ერთობლივი ალბათობა, მაშინ ამ ორ ელემენტარულ შეტყობინებაში არსებული ინფორმაციის რაოდენობა ტოლი იქნება თითოეულ მათგანში არსებული ინფორმაციის რაოდენობათა ჯამისა.

λ_i და λ_j შეტყობინებების ერთდროული გამოჩენის $P(\lambda_i, \lambda_j)$ ალბათობა, როგორც ცნობილია, განისაზღვრება სრული ალბათობის ფორმულით:

$$P(\lambda_i, \lambda_j) = P(\lambda_i) P(\lambda_j/\lambda_i) = P \cdot Q. \quad (1.32)$$

მაშინ, ზემოაღნიშნული მე-2 მოთხოვნის შესაბამისად, უნდა დაკმაყოფილდეს შემდეგი პირობა:

$$\varphi\{P \cdot Q\} = \varphi\{P\} + \varphi\{Q\}. \quad (1.33)$$

ადვილია იმის მიხვედრა, რომ ზემოაღნიშნულ ორ მოთხოვნას აკმაყოფილებს შემდეგი სახის ფუნქცია:

$$J(\lambda_i) = a \log P(\lambda_i). \quad (1.34)$$

ამასთან, როგორც a კოეფიციენტი, ასევე ლოგარითმის ფუძე იქნება შემთხვევითად შერჩეული. თუმცა, ინფორმაციის რაოდენობის დადებითი რიცხვით გამოსახვის მიზნით $a = -1$. ლოგარითმის ფუძედ კი ჩვეულებრივ ირჩევენ 2-ს. მაშინ

$$J(\lambda_i) = -\log_2 P(\lambda_i). \quad (1.35)$$

ინფორმაციის გაზომვის ასეთნაირად განსაზღვრულ ერთეულს უწოდებენ *ინფორმაციის ორობით ერთეულს, ანუ ბიტს*. მაგალითად, ელემენტარული შეტყობინებებიდან ამორჩეული იქნება რომელიმე λ_i შეტყობინება და იგი გადაიცემა $P(\lambda_i) = 1/8$ ალბათობით, მაშინ ამბობენ, რომ მასში არსებობს $\log_2(1/8) = 3$ ბიტი ინფორმაცია.

ზოგჯერ ლოგარითმის ფუძედ ირჩევენ ნატურალურ რიცხვს e . ამ შემთხვევაში ინფორმაციის საზომია ნატურალური ერთეული – *ნატ*.

ერთ ელემენტარულ λ_i შეტყობინებაში თავმოყრილი ინფორმაციის რაოდენობა ჯერ კიდევ სრულად არ ახასიათებს შეტყობინების წყაროს. ელემენტარული შეტყობინების ერთ ჯგუფს შეიძლება გააჩნდეს ბევრი ინფორმაცია, მაგრამ გადაიცემოდეს იშვიათად, სხვა ჯგუფი კი შეიძლება გადაიცემოდეს ხშირად, მაგრამ გააჩნდეს ნაკლები ინფორმაცია. ამიტომ შეტყობინების წყარო შეიძლება დახასიათდეს *ინფორმაციის საშუალო რაოდენობით, რომელიც გადაანგარიშებულია ერთ ელემენტარულ შეტყობინებაზე*. მას უწოდებენ "*წყაროს ენტროპიას*" და იგი განისაზღვრება შემდეგნაირად:

$$H(\lambda) = -\sum_{i=1}^K P(\lambda_i) \cdot \log P(\lambda_i), \quad i=1, K. \quad (1.36)$$

ენტროპია, როგორც წყაროს ინფორმატიულობის რაოდენობრივი საზომი, ხასიათდება შემდეგი თვისებებით:

1. ენტროპია ნივთიერი, შეზღუდული და არაუარყოფითი სიდიდეა. მისი ეს თვისებები გამომდინარეობს $H(\lambda)$ -ს გამოსახულებიდან და, აგრეთვე, იმის გათვალისწინებიდან, რომ $0 < P(\lambda_i) < 1$.

2. დეტერმინირებული შეტყობინებების ენტროპია ნულის ტოლია ($H(\lambda)=0$), თუ თუნდაც ერთ-ერთი შეტყობინების ალბათობაა 1.

3. ენტროპია მაქსიმალურია, თუ λ_i შეტყობინებები თანაბარალობათურია, ანუ თუ $P(\lambda_1)=P(\lambda_2)=\dots=P(\lambda_K)=1/K$. მაშინ

$$H(\lambda)=-\sum_{i=1}^K (1/K) \log(1/K) = \log K. \quad (1.37)$$

როგორც (1.37) ფორმულიდან ჩანს, თანაბარალობათური შეტყობინებების შემთხვევაში ენტროპია იზრდება წყაროს ალფაბეტის მოცულობის (ანუ შეტყობინებათა რაოდენობის) ზრდასთან ერთად. არათანაბარალობათური ელემენტარული λ_i შეტყობინებების შემთხვევაში ენტროპია, შესაბამისად, მცირდება.

4. ორობითი ($K=2$) წყაროს ენტროპია შეიძლება იცვლებოდეს ნულიდან ერთამდე. მართლაც, ორი λ_1 და λ_2 შეტყობინების მაფორმირებელი სისტემის (წყაროს) ენტროპია

$$H(\lambda)=-P(\lambda_1) \cdot \log P(\lambda_1) - P(\lambda_2) \cdot \log P(\lambda_2) = P(\lambda_1) \cdot \log P(\lambda_1) - \{1 - P(\lambda_1)\} \cdot \log \{1 - P(\lambda_1)\}. \quad (1.38)$$

(1.38) ფორმულიდან ჩანს, რომ ენტროპია ნულის ტოლია, თუ $P(\lambda_1)=0$ და $P(\lambda_2)=1$, ან თუ $P(\lambda_1)=1$ და $P(\lambda_2)=0$. ამასთანავე ენტროპია მაქსიმალური, ანუ 1 ბიტის ტოლი იქნება იმ შემთხვევაში, როდესაც $P(\lambda_1)=P(\lambda_2)=1/2$.

1.6.1. რთული შეტყობინებების ენტროპია. წყაროს სიჭარბე

წყაროს ზემოთ განხილული მახასიათებლები (ინფორმაციის რაოდენობა და ენტროპია) განსაზღვრული იყო მხოლოდ ერთ წყაროსათვის, რომელიც გამოიმუშავებს ურთიერთდამოუკიდებელი (მარტივი) შეტყობინებების ნაკადს, ანუ წყაროსათვის მახსოვრობის გარეშე. რეალურ შემთხვევებში წყაროს მიერ გამოიმუშავებული ელემენტარული შეტყობინებების დამოუკიდებლობა საკმაოდ იშვიათი მოვლენაა. უფრო ხშირად კი პირიქით ხდება – არსებობს ძლიერი დეტერმინირებული, ანუ სტატისტიკური კავშირი ერთი ან რამდენიმე წყაროს შეტყობინების ელემენტებს შორის. მაგალითად, ტექსტის გადაცემისას ცალკეული ასოების გამოჩენის ალბათობები დამოკიდებულია მათ გამოჩენამდე

ტექსტში არსებულ (წინმსწრებ) ასოებზე. კერძოდ, ქართული ტექსტისათვის თუ გადაცემულია ასო "ზ", მაშინ მის შემდეგ "ა"-ს გადაცემის ალბათობა გაცილებით მაღალია, ვიდრე "ზ"-სი, ხოლო "ზ"-ს შემდეგ "ჰ" სერთოდ არ გვხვდება ქართულ ტექსტში. მსგავსი სურათი შეინიშნება გამოსახულებათა გადაცემისას, კერძოდ კი გამოსახულების მეზობელი ელემენტები თითქმის ერთნაირი სიკაშკაშისა და ფერისაა.

მონაცემთა გადაცემისა და შენახვისას ასევე ხშირია მრავალი ისეთი წყაროს არსებობა, რომლებიც აფორმირებენ სტატისტიკურად ურთიერთდაკავშირებულ შეტყობინებებს. ასეთი წყაროების მიერ გამომუშავებულ შეტყობინებებს *რთული შეტყობინებები*, ხოლო შესაბამის წყაროებს – *მახსოვრობიან წყაროებს* უწოდებენ.

ნათელია, რომ სტატისტიკურად ურთიერთდაკავშირებული ელემენტებიანი შეტყობინებების ენტროპიისა და ინფორმაციის რაოდენობის განსაზღვრისას არ შეიძლება მხოლოდ უპირობო ალბათობებით შემოფარგვლა. ამ შემთხვევაში აუცილებლად საჭიროა აგრეთვე ცალკეული შეტყობინების გამომჩენის პირობითი ალბათობის გათვალისწინება.

განესაზღვროთ ორი ურთიერთდამოკიდებული წყაროს მიერ გამომუშავებული რთული შეტყობინების ენტროპია (ანალოგიურად განისაზღვრება ერთი მახსოვრობიანი წყაროს მიერ გამომუშავებული რთული შეტყობინების ენტროპია).

დავუშვათ, რომ პირველი წყაროს მიერ გამომუშავებული შეტყობინებები ღებულობენ $x_1, x_2, x_3, \dots, x_k$ მნიშვნელობებს, რომელთა ალბათობებია, შესაბამისად, $P(x_1), P(x_2), P(x_3), \dots, P(x_k)$, ხოლო მეორე წყაროს მიერ გამომუშავებული შეტყობინებებია $y_1, y_2, y_3, \dots, y_m$, რომელთა ალბათობებია $P(y_1), P(y_2), P(y_3), \dots, P(y_m)$.

შეტყობინებათა ორი X და Y წყაროს ერთობლივი ენტროპია შეიძლება განისაზღვროს შემდეგნაირად:

$$H(X, Y) = -\sum_{i=1}^k \sum_{j=1}^m P(x_i, y_j) \log P(x_i, y_j), \quad (1.39)$$

სადაც $P(x_i, y_j)$ x_i და y_j შეტყობინებების ერთდროული გამომჩენის ალბათობაა. ვინაიდან ერთობლივი $P(x_i, y_j)$ ალბათობა ბეისის ფორმულის მიხედვით განისაზღვრება როგორც

$$P(x_i, y_j) = P(x_i) \cdot P(y_j/x_i) = P(y_j) \cdot P(x_i/y_j), \quad (1.40)$$

ამიტომ ერთობლივი ენტროპიის გამოსახულება შეიძლება ჩაიწეროს შემდეგი სახით:

$$\begin{aligned}
H(X,Y) &= -\sum_{i=1}^k \sum_{j=1}^m P(x_i) \cdot P(y_j/x_i) \cdot \log\{P(x_i) \cdot P(y_j/x_i)\} = \\
&= -\sum_{i=1}^k P(x_i) \cdot \log P(x_i) \cdot \sum_{j=1}^m P(y_j/x_i) - \sum_{i=1}^k P(x_i) \cdot \sum_{j=1}^m P(y_j/x_i) \cdot \log P(y_j/x_i). \quad (1.41)
\end{aligned}$$

რადგანაც x_i შეტყობინების გადაცემას აუცილებლად შეესაბამება Y ანსამბლიდან ერთ-ერთი (ნებისმიერი) შეტყობინების გადაცემა, ამიტომ

$$\sum_{j=1}^m P(y_j/x_i) = 1 \quad (1.42)$$

და ერთობლივი $H(X,Y)$ ენტროპია განისაზღვრება შემდეგი ფორმულით:

$$H(X,Y) = -\sum_{i=1}^k P(x_i) \cdot \log P(x_i) - \sum_{i=1}^k P(x_i) \cdot \sum_{j=1}^m P(y_j/x_i) \cdot \log P(y_j/x_i) = H(X) + \sum_{i=1}^k P(x_i) \cdot H(Y/x_i). \quad (1.43)$$

სადაც $H(Y/x_i)$ ეგრეთ წოდებული კერძო პირობითი ენტროპიაა, რომელიც ასახავს Y შეტყობინებას იმ პირობებში, როდესაც არსებობს x_i შეტყობინება.

(1.43) ფორმულის მე-2 შესაკრები წარმოადგენს $H(Y/x_i)$ -ის გასაშუალებას ყველა x_i შეტყობინების მიხედვით და მას Y წყაროს საშუალო პირობითი ენტროპია ეწოდება X შეტყობინების გადაცემის პირობებში. და ბოლოს:

$$H(X,Y) = H(X) + H(Y/X). \quad (1.44)$$

ამრიგად, *ორი შეტყობინების ერთობლივი ენტროპია ტოლია ერთ-ერთი მათგანის უპირობო და მეორის პირობითი ენტროპიების ჯამისა.*

შეიძლება აღინიშნოს რთული შეტყობინებების ენტროპიის ძირითადი თვისებები:

1. *სტატისტიკურად დამოუკიდებელი X და Y შეტყობინებების შემთხვევაში ერთობლივი ენტროპია თითოეული წყაროს ენტროპიათა ჯამის ტოლია:*

$$H(X,Y) = H(X) + H(Y). \quad (1.45)$$

ვინაიდან $H(Y/X) = H(Y)$.

2. X და Y შეტყობინებების *სტატისტიკურად სრულად დამოკიდებულების შემთხვევაში ერთობლივი ენტროპია ერთ-ერთი შეტყობინების უპირობო ენტროპიის ტოლია.* ამასთან მეორე შეტყობინება არ უმატებს ინფორმაციას. მართლაც, შეტყობინებათა სრული სტატისტიკური დამოკიდებულების შემთხვევაში პირობითი $P(y_j/x_i)$ და $P(x_i/y_j)$ ალბათობები ან ნულის ან ერთის ტოლია. მაშინ

$$P(x_i/y_j) \cdot \log P(x_i/y_j) = P(y_j/x_i) \cdot \log P(y_j/x_i) = 0 \quad (1.46)$$

და, შედეგად, $H(X,Y) = H(X) = H(Y)$.

3. პირობითი ენტროპია იცვლება

$$0 < H(Y/X) < H(Y) \quad (1.47)$$

შუალედში.

4. ორი წყაროს ერთობლივი ენტროპიისათვის ყოველთვის სამართლიანია შემდეგი თანაფარდობა:

$$H(X, Y) \leq H(X) + H(Y), \quad (1.48)$$

ამასთანავე ტოლობა სრულდება მხოლოდ შეტყობინებათა დამოუკიდებელი წყაროებისათვის.

ქემოაღნიშნულიდან გამომდინარე, შეიძლება ითქვას, რომ *ელემენტარულ შეტყობინებათა შორის კავშირის არსებობისას წყაროს ენტროპია მცირდება*, თანაც მით უფრო მეტად, რაც უფრო ძლიერია კავშირი შეტყობინებათა ელემენტებს შორის.

ამრიგად, შეტყობინებათა წყაროების ინფორმაციულობის შესახებ შეიძლება გაკეთდეს შემდეგი დასკვნები:

1. წყაროს ენტროპია და ინფორმაციის რაოდენობა მით მეტია, რაც უფრო დიდია წყაროს ალფაბეტის ზომა.

2. წყაროს ენტროპია დამოკიდებულია შეტყობინებათა სტატისტიკურ თვისებებზე. ენტროპია მაქსიმალურია, თუ წყაროს შეტყობინებები თანაბარალობათური და სტატისტიკურად დამოუკიდებელია.

3. იმ წყაროს ენტროპია, რომელიც გამოიმუშავებს არათანაბარალობათურ შეტყობინებებს, ყოველთვის ნაკლებია მაქსიმალურად მიღწევად სიდიდეზე.

4. ელემენტარულ შეტყობინებებს (წყაროს მახსოვრობას) შორის სტატისტიკური კავშირების არსებობისას მისი ენტროპია მცირდება.

მაგალითისათვის განვიხილოთ წყარო, რომელიც გამოიმუშავებს ქართული ალფაბეტის *ა, ბ, გ, ... , ჰ* ასოებს, რომელთა რაოდენობა, როგორც ცნობილია, 33-ია. სიმარტივის მიზნით ჩავთვალოთ, რომ მათი საერთო რაოდენობა (ალფაბეტის ზომა) $K=25=32$.

ქართული ალფაბეტის ყველა ასონიშანს რომ ერთნაირი ალბათობა ქონდეს და ისინი სტატისტიკურად დამოუკიდებელი რომ იყოს, მაშინ ერთ სიმბოლოზე გადაანგარიშებული საშუალო ენტროპია

$$H(\lambda)_{max} = \log_2 32 = 5 \text{ ბიტი/ასონიშანზე} .$$

თუ ახლა გავითვალისწინებთ ტექსტში ასოების გამოჩენის სხვადასხვა ალბათობას, მაშინ საანგარიშო ენტროპიაა

$$H(\lambda) = 4,39 \text{ ბიტი/ასონიშანზე} .$$

ორ და სამ მეზობელ ასონიშნებს შორის კორელაციის (სტატისტიკური კავშირის) გათვალისწინებით ენტროპია მცირდება, შესაბამისად

$$H(\lambda)=3,52 \text{ ბიტი/ასონიშნამდე} \text{ და } H(\lambda)=3,05 \text{ ბიტი/ასონიშნამდე} .$$

თუ კორელაციას გავითვალისწინებთ რვა და უფრო მეტი რაოდენობის ასონიშნებს შორის, მაშინ ენტროპია შემცირდება

$$H(\lambda)=2,0 \text{ ბიტი/ასონიშნამდე}$$

და შემდეგ რჩება უცვლელი.

იმასთან დაკავშირებით, რომ ერთი და იმავე ზომის მქონე ალფაბეტის რეალურ წყაროებს შეიძლება ჰქონდეს სრულიად განსხვავებული ენტროპია (ეს კი არა მარტო ტექსტია, არამედ შეიძლება იყოს სიტყვა, მუსიკა, გამოსახულებები და ა.შ.), შემოაქვტ წყაროს ისეთი მახასიათებელი, როგორცაა სიჭარბე

$$\rho_{\text{წყ.}}=1-H(\lambda)/H(\lambda)_{\text{max}}=1-H(\lambda)/\log K ,$$

სადაც $H(\lambda)$ რეალური წყაროს ენტროპიაა, $\log K$ მაქსიმალურად მიღწევადი ენტროპიაა იმ წყაროსათვის, რომლის ალფაბეტის რაოდენობაა K . მაშინ ლიტერატურული ტექსტის სიჭარბე, მაგალითად, ტოლი იქნება:

$$\rho_{\text{წყ.}}=1-(2 \text{ ბიტი/ასონიშნზე})/(5 \text{ ბიტი/ასონიშნზე})=0,6 .$$

სხვა სიტყვებით რომ ვთქვათ, კავშირის არხით ტექსტის გადაცემისას ათი გადაცემულიდან ყოველი ექვსი ასონიშანი არ ატარებს არავითარ ინფორმაციას და ამიტომ ისინი შეიძლება არ გადაიცეს ყოველგვარი დანაკარგის გარეშე.

ასეთივე, თუ უფრო მეტი ($\rho_{\text{წყ.}}=0,9...0,95$) არა სიჭარბე გააჩნია ინფორმაციის სხვა (სიტყვა, განსაკუთრებით კი მუსიკა, სატელევიზიო გამოსახულებები და სხვა) წყაროებს.

ზემოაღნიშნულიდან გამომდინარე, ისმის სრულიად სამართლიანი შეკითხვა: საჭიროა კი დავაკავოთ ინფორმაციის მატარებელი ან კავშირის არხი იმ სიმბოლოების გადაცემით, რომლებსაც პრაქტიკულად არავითარი ინფორმაცია არ გადააქვთ, ან კიდევ შესაძლებელია თუ არა საწყისი შეტყობინების ისეთი გარდაქმნა, რომ ინფორმაცია "ჩაეტოს" ამისათვის საჭირო სიმბოლოების მინიმალურ რაოდენობაში?

აღმოჩნდა, რომ არა მარტო შესაძლებელია დასმულ კითხვაზე დადებითი პასუხის არსებობა, არამედ აუცილებელიცაა. დღეისათვის ინფორმაციის გადაცემისა და კავშირის მრავალი არსებული რადიოტექნიკური სისტემა უბრალოდ ვერ იმუშავებდა, მათში რომ არ ხორციელდებოდა აღნიშნული სახის კოდირება. არ იარსებებდა **GSM** და **GDMA** სტანდარტების ციფრული ფიჭური

კავშირი, არ იმუშავებდა ციფრული თანამგზავრული ტელეხედვის სისტემები, ძალზე არაეფექტური იქნებოდა **Internet**-ის მუშაობა, ხოლო რაც შეეხება ლაზერული დისკიდან ვიდეოფილმის ნახვას ან კარგი მუსიკის მოსმენას, ამაზე ლაპარაკიც კი ვერ იქნებოდა. ყოველივე ეს უზრუნველყოფილი ხდება აღნიშნულ სისტემებში ინფორმაციის ეფექტური, ანუ ეკონომიური კოდირების რეალიზაციით.

თანამედროვე რადიოტექნიკის აღნიშნული მიმართულების (ეკონომიური, ანუ უჭარბო კოდირების თეორიისა და ტექნიკის საფუძვლების) შესწავლას ეთმობა წინამდებარე კურსის შემდეგი ნაწილი.

2. ეკონომიური კოდირების საფუძვლები

ინფორმაციის გადაცემის რადიოტექნიკური სისტემის გამოყენებით გადასაცემი შეტყობინებები (სიტყვა, მუსიკა, სატელევიზიო გამოსახულებები და ა.შ.) უმთავრესად განკუთვნილია ადამიანის მგრძობელობის ორგანოების აღქმისათვის და ისინი ჩვეულებრივ ცუდადაა თავსებადი კავშირის არხის საშუალებით მათი ეფექტური გადაცემისადმი. ამიტომ გადაცემის პროცესში ისინი,

როგორც წესი, ექვემდებარებიან კოდირებას.

რა არის კოდირება და რატომ იყენებენ მას? საზოგადოდ *კოდირებაში* იგულისხმება შეტყობინების $A\{x_i\}$, ($i = 1, 2, \dots, K$) ალფაბეტის გარდაქმნა გარკვეული სახით შერჩეული კოდური სიმბოლოების $\mathcal{R}\{x_j\}$, ($j = 1, 2, \dots, N$) ალფაბეტად. ჩვეულებრივ (მაგრამ არა აუვილებლად) კოდური სიმბოლოების ალფაბეტის ზომა $\dim \mathcal{R}\{x_j\}$ ნაკლებია ან ბევრად ნაკლებია წყაროს ალფაბეტის $\dim A\{x_i\}$ ზომაზე.

შეტყობინებათა კოდირებას შეიძლება გააჩნდეს სხვადასხვა მიზანი. მაგალითად, ეს შეიძლება იყოს კოდირება ინფორმაციის გასაიდუმლოების მიზნით. ამასთან $A\{x_i\}$ ალფაბეტის ელემენტარულ x_i შეტყობინებებს შეუსაბამებენ, მაგალითად, იმ სპეციალური კოდური ცხრილების ციფრების ან ასოების მიმდევრობებს, რომლებიც ცნობილია მხოლოდ ინფორმაციის გამგზავნისა და მიმღების მიერ.

კოდირების მეორე მაგალითად შეიძლება მოვიყვანოთ დისკრეტულ x_i შეტყობინებათა გარდაქმნა ათვლის ერთი სისტემიდან მეორეში (ათობითი სისტემიდან ორობითში, რვაობითში და ა.შ., არაპოზიციურიდან პოზიციურში, ასოების ალფაბეტის ციფრულში გადაყვანა და სხვა.)

ინფორმაციის საარხო კოდირება, ანუ ხელშეშლამდგრადი კოდირება შეიძლება გამოყენებული იყოს ხელშეშლებიანი არხით გადაცემისას წარმოქმნილ შეცდომათა რაოდენობის შემცირებისათვის.

და ბოლოს, შეტყობინებათა კოდირება შეიძლება განხორციელდეს ინფორმაციის მოცულობის შემცირებისა და მისი გადაცემის სიჩქარის გაზრდის ან გადაცემისათვის საჭირო სიხშირეთა ზოლის შემცირების მიზნით. ასეთ კოდირებას უწოდებენ *ეკონომიურ*, *უჭარბო* ან *ეფექტურ კოდირებას*. კურსის მოცემულ ნაწილში სწორედ ასეთი სახის კოდირება იქნება განხილული. ჩვეულებრივ კოდირების პროცესს წინ უძღვის (და მასშია ჩართული) უწყვეტი

$\lambda(t)$ შეტყობინების დისკრეტიზაცია, და დაკვანტვა, ანუ მისი გარდაქმნა ელემენტარული $\{\lambda_i\}$ დისკრეტული შეტყობინებების მიმდევრობად.

მანამ, სანამ გადავიდოდეთ *ეკონომიური კოდირების* საკითხზე, მოკლედ გავარჩიოთ თავად კოდირების პროცედურის არსი.

წყაროს K ტევადობის $A\{\lambda_i\}$ ალფაბეტის ნებისმიერი დისკრეტული λ_i შეტყობინება შეიძლება კოდირებული იყოს $\mathcal{R}\{x_j\}$ ალფაბეტის შესაბამისი წესით შერჩეული კოდური x_j სიმბოლოების მიმდევრობით.

მაგალითად, ნებისმიერი რიცხვი (ხოლო λ_i შეიძლება მივიჩნიოთ რიცხვად) შეიძლება ათვლის მოცემულ პოზიციურ სისტემაში ჩაიწეროს ჩაიწეროს შემდეგი სახით:

$$\lambda_i = M = x_{n-1} \cdot m^{n-1} + x_{n-2} \cdot m^{n-2} + \dots + x_0 \cdot m^0, \quad (2.1)$$

სადაც: m ათვლის სისტემის ფუძეა; $x_0 \dots x_{n-1}$ m -ის ხარისხების წინ არსებული კოეფიციენტებია; $x \in \{0, m-1\}$.

ვთქვათ, მაგალითად, $\lambda_i = M = 59$. მაშინ $m = 8$ -ის შემთხვევაში მის კოდს ექნება სახე

$$M = 59 = 7 \cdot 8^1 + 3 \cdot 8^0 = 73_8.$$

თუ $m = 4$, მაშინ:

$$M = 59 = 3 \cdot 4^2 + 2 \cdot 4^1 + 3 \cdot 4^0 = 323_4.$$

და ბოლოს, იმ შემთხვევაში, როდესაც $m = 2$, მაშინ

$$M = 59 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 111011_2.$$

ამრიგად, რიცხვები 73 , 323 და 111011 შესაბამისად შეიძლება ჩაითვალოს $M = 59$ -ის რვაობით, ოთხობით და ორობით კოდებად.

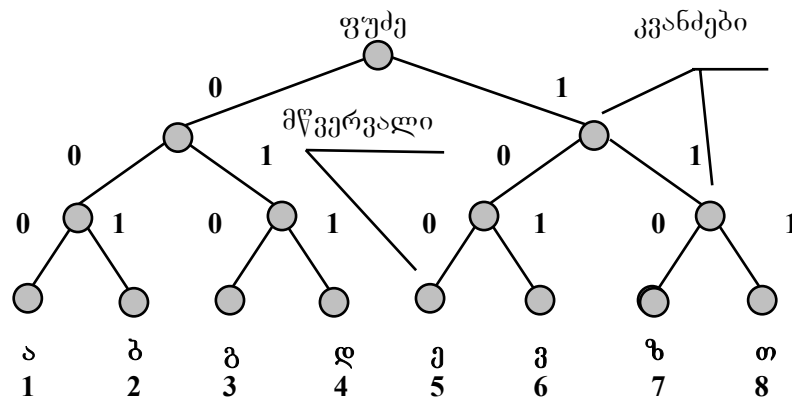
კოდის ფუძე, პრინციპში, შეიძლება ნებისმიერი იყოს, თუმცა ყველაზე მეტად მაინც *ორობითი კოდება* (ანუ კოდები 2-ის ფიძით) გავრცელებული, რომლებსთვისაც კოდური $\mathcal{R}\{x_j\}$ სიმბოლოების ალფაბეტის ზომა ორის ტოლია ($x_j \in \{0,1\}$). ორობითი, ანუ მხოლოდ *ნულიანებისა და ერთიანების* შემცველი კოდები მარტივად ფორმირდება და გადაიცემა კავშირის არხებით და, რაც მთავარია, ისინი წარმოადგენს ციფრული ელექტრონული გამომთვლელი მანქანების შიდა სამუშაო ენას, რაც იმას ნიშნავს, რომ ისინი ყოველგვარი გარდაქმნების გარეშე შეიძლება დამუშავდეს ციფრული საშუალებებით. ამიტომ, იმ დროს, როდესაც საუბარი ეხება კოდირებას ან კოდებს, უმთავრესად მხედველობაში აქვთ სწორედ *ორობითი კოდები*. ამიტომაც მომავალში

ძირითადად განვიხილავთ ორობით კოდირებას. კოდების წარმოდგენის ან მოცემის უმარტივეს ხერხს წარმოადგენს კოდური ცხრილები, რომლებიც გვიჩვენებენ შესაბამისობას λ_i შეტყობინებებსა და შესაბამის კოდებს შორის (ცხრილი 2.1).

ცხრილი 2.1

ასო λ_i	რიცხვი λ_i	კოდი 10-ის ფუძით	კოდი 4-ის ფუძით	კოდი 2-ის ფუძით
ა	0	0	00	000
ბ	1	1	01	001
გ	2	2	02	010
დ	3	3	03	011
ე	4	4	10	100
ვ	5	5	11	101
ზ	6	6	12	110
თ	7	7	13	111

კოდების აღწერის თვალნათელი და მოხერხებული მეთოდია მათი წარმოდგენა კოდური ხის სახით (ნახ. 2.1).



ნახ. 2.1. თანაბარი სამთანრიგა კოდის წარმოდგენა კოდური ხის გამოყენებით

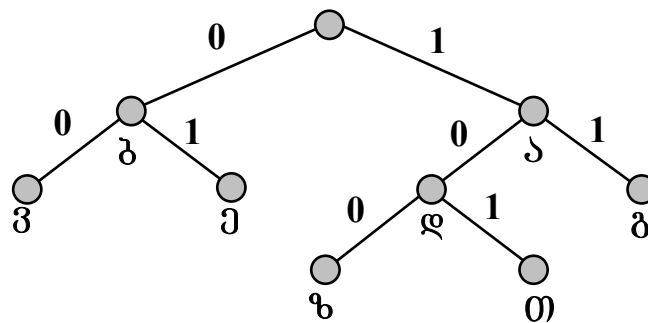
რომელიმე წერტილიდან (კოდური ხის ფუძიდან) მოცემული კოდისათვის კოდური ხის ასაგებად ატარებენ შტოებს (0 ან 1). კოდური ხის მწვერვალებზე პოულობენ წყაროს აღფაბეტის ასოებს, თანაც ყოველ ასონიშანს შეუსაბამებენ თავის მწვერვალს და თავის გზას ფუძიდან მწვერვალისაკენ. მაგალითად, ასონიშან ა-ს შეესაბამება კოდი 000, ასონიშან ვ-ს – 101 და ა.შ.

2.1 ნახაზზე გამოსახული კოდური ხით მიღებულ კოდი წარმოადგენს თანაბარ სამთანრიგა კოდს.

თანაბარი კოდები ფართოდ გამოიყენება თავისი სიმარტივისა და კოდირება-დეკოდირების პროცესის მოხერხებულობის გამო, რაც გამოიხატება შემდეგში: ყოველ ასონიშანს შეესაბამება ბიტების ერთნაირი რაოდენობა; ბიტების მოცემული რაოდენობის მიხედვით კოდური ცხრილი იძლევა შესაბამისი ასონიშნის განსაზღვრის საშუალებას.

თანაბარ კოდებთან ერთად შეიძლება გამოყენებული იყოს არათანაბარი კოდებიც. ამ შემთხვევაში წყაროს ალფაბეტის ყოველი ასონიშნის კოდირება ხდება სიმბოლოთა სხვადასხვა რაოდენობით. მაგალითად, ა – 10, ბ – 110, გ – 1110 და ა.შ.

არათანაბარი კოდირებისათვის კოდურ ხეს შეიძლება ქონდეს 2.2 ნახაზზე ნაჩვენები სახე.



ნახ. 2.2. არათანაბარი კოდირების მაილუსტრირებელი კოდური ხე

აღნიშნული კოდის გამოყენებისას ასონიშან ა-ს შესაბამისი კოდია 1, ასონიშან ბ-სი – 0, ბ-სი – 11 და ა.შ. თუმცა, შეიძლება შევნიშნოთ, რომ მაგალითად ტექსტის აბბა 1001-ით კოდირებისას ჩვენ ვერ შევძლებთ მის დეკოდირებას ცალსახად, ვინაიდან ასეთივე კოდს იძლევა შემდეგი ფრაზები: ზა, აზა და ჯე. ასეთი კოდები, რომლებიც ვერ უზრუნველყოფენ ცალსახა დეკოდირებას, დაყვანადი ან არაპრეფიქსული კოდები ეწოდებათ. პრაქტიკაში ისინი შეუძლებელია გამოყენებული იყოს სპეციალური გამყოფი სიმბოლოების გარეშე. ასეთი ტიპის კოდების გამოყენების მაგალითია მორზეს ანბანი, რომელშიც წერტილებისა და ტირეების გარდა არსებობს ასონიშნების გამომყოფი სპეციალური სიმბოლოები და სიტყვები. მაგრამ ეს უკვე არაა ორობითი კოდი.

თუმცა შეიძლება აიგოს არათანაბარი არადაყვანადი კოდები, რომელთა საშუალებით შესაძლებელია ცალსახა დეკოდირება. ამისათვის საჭიროა, რომ

ვთქვათ, შეტყობინების წყარო გასცემს ერთნაირი ალბათობის 8-დან 0-მდე 8 ასონიშნისაგან ($P(\lambda_i)=1/8$) შედგენილ რაღაც ტექსტს. კოდერი ამ ასონიშნებს შეუსაბამებს თანაბარ სამთანრივ კოდს (იხ. ცხრილი 2.1).

განვსაზღვროთ ასეთი ალფაბეტის მაფორმირებელი წყაროს ძირითადი ინფორმაციული მახასიათებლები:

– წყაროს ენტროპია

$$H(\bar{\lambda}) = -\sum_{i=1}^K P_i \log P_i, H(\bar{\lambda}) = \log K = 3;$$

– წყაროს სიჭარბე

$$\rho_u = 1 - \frac{H(\bar{\lambda})}{\log K} = 0;$$

– კოდის სიმბოლოთა საშუალო რაოდენობა

$$\bar{n} = \sum_{i=1}^K n_i \cdot P_i = \sum_{i=1}^8 3 \cdot \frac{1}{8} = 3;$$

– კოდის სიჭარბე

$$\rho_k = 1 - \frac{H(\bar{\lambda})}{\bar{n}} = 0.$$

ამრიგად, თანაბარალბათური ასონიშნებიანი შეტყობინებების კოდირებისას შერჩეული (თანაბარი) კოდის სიჭარბე აღმოჩნდა ნულის ტოლი.

ამჯერად დავუშვათ, რომ ტექსტში სხვადასხვა ასონიშნების გამოჩენის ალბათობები სხვადასხვაა (ცხრილი 2.2).

ცხრილი 2.2

ა	ბ	გ	დ	ე	ვ	ზ	თ
$P_a=0.6$	$P_b=0.2$	$P_g=0.1$	$P_d=0.04$	$P_e=0.025$	$P_v=0.015$	$P_z=0.01$	$P_t=0.01$

ამ შემთხვევაში წყაროს ენტროპია

$$H(\bar{\lambda}) = -\sum_{i=1}^K P_i \log P_i$$

იქნება ნაკლები და იგი $H(\bar{\lambda}) = 1.781$ -ის ტოლია.

აღნიშნული თანაბარი სამთანრიგა კოდის გამოყენებისას ერთ შეტყობინებაზე გადაანგარიშებული სიმბოლოების საშუალო რაოდენობა

$$\bar{n} = \sum_{i=1}^K n_i P_i = n \sum_{i=1}^K P_i = n = 3.$$

მაშინ კოდის სიჭარბე იქნება

$$\rho_k = 1 - \frac{H(\bar{\lambda})}{\bar{n}} = 1 - \frac{1.781}{3} \approx 0.41,$$

ანუ იგი საკმაოდ მნიშვნელოვანი სიდიდისაა (10 სიმბოლოდან საშუალოდ 4 სიმბოლოს არ გადააქვს არავითარი ინფორმაცია).

იმის გამო, რომ არათანაბარალბათური შეტყობინებების კოდირებისას თანაბარ კოდებს გააჩნია დიდი სიჭარბე, გადაწყვიტეს არათანაბარი კოდების გამოყენება, რომელთა კოდური კომბინაციების ხანგრძლივობა შეთანხმებული იქნებოდა სხვადასხვა ასონიშნის გამოჩენის ალბათობასთან.

ასეთ კოდირებას სტატისტიკურ კოდირებას უწოდებენ.

სტატისტიკური კოდირებისას არათანაბარ კოდებს ისეთნაირად ირჩევენ, რომ უფრო ალბათური ასონიშნები გადაიცეს კოდის უფრო მოკლე კომბინაციებით, ხოლო ნაკლებ ალბათური – უფრო გრძელით. ამის შედეგად მცირდება კოდური ჯგუფის საშუალო სიგრძე თანაბარი კოდირების შემთხვევასთან შედარებით.

ასეთი კოდირების ერთ-ერთ მეთოდი შემოთავაზებულია ხაფმენის მიერ. ხაფმენის არათანაბარი კოდის კოდური ხის აგება სხვადასხვა ალბათობის მქონე რვა λ_i შეტყობინებიდან ერთის გადაცემისათვის წარმოდგენილია 2.3 ცხრილში.

ასონი-შანი	აღბათობა P_i	კოდური ხე	კოდი	n_i	$n_i \cdot P_i$
კ	0.6		1	1	0.6
ბ	0.2		01	2	0.4
გ	0.1		001	3	0.3
დ	0.04		0001	4	0.16
ე	0.025		00001	5	0.125
ვ	0.015		000001	6	0.08
ზ	0.01		0000001	7	0.07
თ	0.01		00000001	8	0.08

აღნიშნული კოდისათვის სიმბოლოთა საშუალო რაოდენობაა

$$\bar{n} = \sum_{i=1}^8 n_i P_i = 1.825,$$

ხოლო კოდის სიჭარბე

$$\rho_k = 1 - \frac{H(\bar{\lambda})}{\bar{n}} = 1 - \frac{1.781}{1.825} \approx 0.03,$$

ანუ იგი ერთი რიგით ნაკლებია თანაბარ კოდირებასთან შედარებით.

სტატისტიკური კოდირების მეორე უმარტივეს ხერხს წარმოადგენს კოდირება შენონ-ფანოს მეთოდით. ამ აღგორითმის მიხედვით კოდირება ხორციელდება შემდეგნაირად:

- თავდაპირველად შეტყობინების აღფაბეტის ყველა ასონიშანს ჩაწერენ მათი აღბათობის კლებადობის ნიშნით;
- შემდეგ ასონიშნების მთლიან ერთობლიობას დაყოფენ აღბათობათა ჯამის მიხედვით დაახლოებით ტოლ ორ ჯგუფად; ერთ-ერთ მათგანს (ჯგუფში შეიძლება იყოს სიმბოლოთა ნებისმიერი რაოდენობა, მათ შორის – ერთი) ანიჭებენ სიმბოლოს "1", ხოლო მეორეს – სიმბოლოს "0";
- აღნიშნული ჯგუფებიდან თითოეულ მათგანს კვლავ ყოფენ (თუ ეს შესაძლებელია) ორ ნაწილად, რომელთაგან თითოეულ მათგანს ანიჭებენ "1"-ს და "0"-ს და ა.შ.

შენონ-ფანოს მეთოდით კოდირების პროცედურა ილუსტრირებულია 2.4 ცხრილში.

ასონი- შანი	$P(\lambda_i)$	I	II	III	IV	V	კოდი	$n_i \cdot P_i$
ა	0.6	1					1	0.6
ბ	0.2	0	1	1			011	0.6
გ	0.1			0			010	0.3
დ	0.04		0	1			001	0.12
ე	0.025			0	0	1		0001
ვ	0.015		0			0	0	
ზ	0.01				1			
თ	0.01					0	000000	0.06

ასეთნაირად მიღებული კოდისათვის ერთ ასონიშანზე გადაანგარიშებული ორობითი სიმბოლოების საშუალო რაოდენობა ტოლია

$$\bar{n} = \sum_{i=0}^7 n_i P_i \approx 1.9,$$

ხოლო კოდის სიჭარბეა

$$\rho_k = 1 - \frac{1.781}{1.9} \approx 0.06,$$

ანუ ის გაცილებით ნაკლებია თანაბარ კოდირებასთან შედარებით.

ყურადღება მივაქციოთ იმ ფაქტს, რომ როგორც ხაფშენის, ასევე შენონ-ფანოს კოდისათვის წყაროს ერთ სიმბოლოზე გადაანგარიშებული ორობითი სიმბოლოების საშუალო რაოდენობა უახლოვდება წყაროს ენტროპიას, მაგრამ მაინც არაა მისი ტოლი. ეს არის შედეგი უხმაურო კოდირების თეორემისა (შენონის პირველი თეორემისა) წყაროსათვის, რომელიც ამტკიცებს შემდეგს:

ნებისმიერი წყაროს კოდირება შესაძლებელია ორობითი მიმდევრობით წყაროს λ_i სიმბოლოზე გადაანგარიშებული ორობითი სიმბოლოების ისეთი რაოდენობისას, რომელიც რაც შეიძლება მეტად ახლოა ენტროპიასთან, და შეუძლებელია კოდის ისეთი სიგრძის მიღწევა, რომელიც ნაკლებია $H(\lambda)$ ენტროპიაზე.

ძნელია ამ თეორემის გადაჭარბებული შეფასება თანამედროვე რადიოტექნიკისათვის – იგი ადგენს ინფორმაციის კომპაქტური წარმოდგენის საზღვრებს, რომელთა მიღწევა შესაძლებელია მისი სწორი კოდირებისას.

მონაცემების აღდგენა. დეკოდერის მიერ ფორმირებული აღდგენილი მონაცემები შეიძლება ან აბსოლუტურად ზუსტად დაემთხვეს წყაროს საწყის მონაცემებს, ან უმნიშვნელოდ განსხვავდებოდეს მათგან.

არსებობს მონაცემთა კომპრესიის სისტემების ორი ტიპი:

1. კომპრესიის (შეკუმშვის) სისტემები ინფორმაციის დანაკარგების გარეშე (დაურღვევადი კომპრესია);

2. კომპრესიის (შეკუმშვის) სისტემები ინფორმაციის დანაკარგებით (რღვევადი კომპრესია).

2.1.1. კომპრესია ინფორმაციის დანაკარგების გარეშე

დანაკარგების გარეშე კომპრესიის სისტემებში დეკოდერი წყაროს მონაცემებს აღადგენს აბსოლუტურად ზუსტად, რის გამოც კომპრესიის (შეკუმშვის) სისტემის სტრუქტურას შემდეგი სახე ექნება:

მონაცემთა ვექტორები $X \rightarrow$ კოდერი $\rightarrow B(X) \rightarrow$ დეკოდერი $\rightarrow X$

წყაროს იმ მონაცემების ვექტორი X , რომელიც ექვემდებარება კომპრესიას, წარმოადგენს სასრულო სიგრძის $X = (x_1, x_2, \dots, x_n)$ მიმდევრობას. x_i ანათვლები, რომლებიც ქმნიან X ვექტორს, შეირჩევა მონაცემთა სასრულო A ალფაბეტიდან. ამასთანავე მონაცემთა ვექტორის ზომა n შეზღუდულია, მაგრამ იგი შეიძლება იყოს რაც შეიძლება დიდი. ამრიგად, წყარო თავის გამოსასვლელზე მონაცემების სახით აფორმირებს A ალფაბეტის X მიმდევრობას, რომლის სიგრძეა n .

კოდერის გამოსასვლელზე მიღებული კომპრესირებული მონაცემები, რომლებიც შემავალი X ვექტორის შესაბამისია, წარმოვადგინოთ ორობითი $B(X) = (b_1, b_2, \dots, b_k)$ მიმდევრობის სახით, რომლის ზომა k დამოკიდებულია X -ზე.

$B(X)$ -ს ვუწოდოთ კოდური სიტყვა, რომელსაც კოდერი ანიჭებს X ვექტორს (ან ვუწოდოთ კოდური სიტყვა, რომელშიაც კოდერი გარდაქმნის X ვექტორს). ვინაიდან კომპრესიის სისტემა დაურღვევადია, ამიტომ ერთნაირ $X_l = X_m$ ვექტორებს უნდა შეესაბამებოდეს ერთნაირი კოდური სიტყვები $B(X_l) = B(X_m)$.

კომპრესიის განხორციელებისას ბუნებრივად წარმოიშევა კითხვა, რამდენად ეფექტურია კომპრესიის (შეკუმშვის) ესა თუ ის სისტემა. ვინაიდან, როგორც ცნობილია, ძირითადად გამოიყენება მხოლოდ ორობითი კოდირება,

ამიტომ ასეთ საზომს შეიძლება წარმოადგენდეს კომპრესიის კოეფიციენტი r , რომელიც განისაზღვრება შემდეგი ფარდობის სახით:

$$r = \frac{\text{წყაროს მონაცემთა ზომა (ბიტებში)}}{\text{კომპრესირებულ მონაცემთა ზომა (ბიტებში)}} = \frac{n \log_2 (\dim A)}{k}, \quad (2.2)$$

სადაც $\dim A$ მონაცემების A ალფაბეტის ზომაა.

ამრიგად, კომპრესიის კოეფიციენტი $r = 2$ ნიშნავს იმას, რომ კომპრესირებულ მონაცემთა მოცულობა ორჯერ ნაკლებია წყაროს მონაცემთა მოცულობასთან შედარებით. რაც უფრო მეტია კომპრესიის r კოეფიციენტი, მით უკეთ მუშაობს მონაცემთა კომპრესიის სისტემა.

კომპრესიის r კოეფიციენტთან ერთად კომპრესიის (შეკუმშვის) სისტემის ეფექტურობა შეიძლება დახასიათდეს კომპრესიის სიჩქარით R

$$R = k/n \quad (2.3)$$

და "წყაროს მონაცემების ერთ ანათვალზე გადაანგარიშებული კოდური ბიტების რაოდენობით". სისტემა, რომელსაც გააჩნია კომპრესიის დიდი კოეფიციენტი, უზრუნველყოფს კომპრესიის ნაკლებ სიჩქარეს.

2.1.2. კომპრესია ინფორმაციის დანაკარგებით

კომპრესიის სისტემაში ინფორმაციის დანაკარგებით (ანუ რღვევად სისტემაში) კოდირება ხორციელდება ისეთნაირად, რომ დეკოდერს არ შეუძლია წყაროს მონაცემების აღდგენა საწყისი სახით. შესაბამისი სისტემის სტრუქტურულ სქემას აქვს შემდეგი სახე:

$$X \rightarrow \text{დამკვანტავი} \rightarrow X^q \rightarrow \text{არარღვევადი კოდერი} \rightarrow B(X^q) \rightarrow \text{დეკოდერი} \rightarrow X^*$$

ისევე როგორც წინა სქემაში, ამ შემთხვევაშიც $X = (x_1, x_2, \dots, x_n)$ საწყის მონაცემთა ვექტორია, რომელიც განიცდის კომპრესიას, ხოლო $X^* = (x_1, x_2, \dots, x_n)$ აღდგენილი ვექტორია. ამ სქემაში ჩართულია კომპრესიის ელემენტი – დამკვანტავი, რომელიც არ არსებობს არარღვევადი კომპრესიის სქემაში.

დამკვანტავი საწყისი მონაცემების X ვექტორისაგან აფორმირებს X^q ვექტორს, რომელიც საშუალოკვადრატული მანძილის (ცდომილების) თვალსაზრისით ახლოა X ვექტორთან. დამკვანტავის მუშაობა ემყარება ალფაბეტის ზომის შემცირებას (უმარტივესი დამკვანტავი ახდენს მონაცემთა დამრგვალებას უახლოეს მთელ რიცხვამდე).

შემდეგ კოდერის საშუალებით ხდება დაკვანტული მონაცემების X^q ვექტორის არარღვევადი კომპრესია ისე, რომ უზრუნველყოფილი ხდება ცალსახა შესაბამისობა X^q სს და $B(X^q)$ -ს შორის ($X_l^q = X_m^q$ -სათვის სრულდება პირობა $B(X_l^q) = B(X_m^q)$). თუმცა მთლიანობაში სისტემა რჩება რღვევადი, ვინაიდან ორ სხვადასხვა X ვექტორს შეიძლება შეესაბამებოდეს ერთი და იგივე ვექტორი X^* . რღვევადი კოდერი ხასიათდება ორი მახასიათებლით – კომპრესიის R სიჩქარითა და დამახინჯებათა D სიდიდით, რომლებიც განისაზღვრება როგორც

$$R = k/n,$$

$$D = (1/n) \sum (x_i - x^*)^2. \quad (2.4)$$

R პარამეტრი ახასიათებს წყაროს ერთ ანათვალზე გადაანგარიშებულ კომპრესიის სიჩქარეს ბიტებში, D კი წარმოადგენს საშუალოკვადრატულ განსხვავებას X^* და X შორის. თუ ერთმანეთს შევადარებთ ორ რღვევად კომპრესიის განმახორციელებელ სისტემას, რომელთა სიჩქარეები და დამახინჯებათა სიდიდეებია შესაბამისად R_1, D_1 და R_2, D_2 , მაშინ პირველი მათგანი უკეთესია, თუ $R_1 < R_2$ და $D_1 < D_2$. თუმცა, სამწუხაროდ, შეუძლებელია რღვევადი კომპრესიის ისეთი სისტემის აგება, რომელიც შესაძლებელს გახდის ერთდროულად R სიჩქარისა და D დამახინჯებათა შემცირებას, ვინაიდან ეს ორი პარამეტრი ერთმანეთის უკუპროპორციულია. ამიტომ დანაკარგებიანი კომპრესიის სისტემის მიზანი შეიძლება იყოს ან სიჩქარის მინიმიზაცია დამახინჯებათა მოცემული მნიშვნელობისათვის, ან უმცირესი შეცდომების მიღება კომპრესიის მოცემული სიჩქარისას.

მონაცემთა არარღვევადი ან რღვევადი კომპრესიის სისტემის შერჩევა საწყისი შესაკუმში მონაცემების ტიპზეა დამოკიდებული. ტექსტური მონაცემების, კომპიუტერული პროგრამების, დოკუმენტების, ნახაზებისა და ა.შ. კომპრესიისას სრულიად ნათელია, რომ გამოყენებული უნდა იყოს არარღვევადი მეთოდები, ვინაიდან აუცილებელია საწყისი მონაცემების აღდგენა აბსოლუტური სიზუსტით. სიტყვის, მუსიკალური მონაცემებისა და გამოსახულებათა კომპრესიისას კი პირიქით, ხშირად იყენებენ რღვევად კომპრესიას, ვინაიდან პრაქტიკულად შეუმჩნეველი დამახინჯებების შემთხვევაში ის უზრუნველყოფს R სიჩქარის ნაკლებობას მთელი რიგით, ზოგჯერ ორი რიგითაც კი. ზოგადად რღვევადი კოდირება, როგორც წესი, უზრუნველყოფს კომპრესიის კოეფიციენტის გაცილებით მნიშვნელოვან სიდიდეს არარღვევად კომპრესიასთან შედარებით.

ქვემოთ მოყვანილია რამდენიმე მაგალითი, რომელთა საშუალებითაც ნათელი ეფინება კომპრესიის პროცედურის განხორციელების აუცილებლობას და ილუსტრირებულია ეკონომიური კოდირების უმარტივესი მეთოდები და მონაცემთა კომპრესიის ეფექტურობა.

მაგალითი 1. დაუშვათ, რომ წყარო გენერირებს 512×512 ელემენტის ციფრულ გამოსახულებას (კადრს), რომელიც შეიცავს 256 ფერს. თითოეული ფერი წაქრმოადგენს $\{0,1,2,\dots, 255\}$ სიმრავლის ერთ-ერთ რიცხვს. მათემატიკურად ეს გამოსახულება წარმოადგენს 512×512 ზომის მატრიცას, რომლის თითოეული ელემენტი ეკუთვნის $\{0,1,2,\dots, 255\}$ სიმრავლეს (გამოსახულების ელემენტებს პიქსელებსაც უწოდებენ). თავის მხრივ $\{0,1,2,\dots, 255\}$ სიმრავლის თითოეული პიქსელი შეიძლება წარმოდგენილი იყოს ორობით ფორმაში 8 ბიტის გამოყენებით. ამრიგად, წყაროს ზომა ბიტებში შეადგენს $8 \times 512 \times 512 = 2^{21}$ მეგაბიტს (მგბიტი). ამ შემთხვევაში ერთ გიგაბაიტთან მყარ დისკზე მოთავსდება დაახლოებით შეუკუმშავი გამოსახულების 5000 კადრი (დაახლოებით 5-წუთიანი ვიდეორგოლი). თუ მოვახდენთ ამ გამოსახულების 10-ჯერად (კომპრესიის კოეფიციენტი $r=10$), მაშინ იმავე დისკზე უკვე შევძლებთ თითქმის ერთსაათიანი ვიდეოფილმის განთავსებას!

დაუშვათ, აგრეთვე, რომ ჩვენ გვინდა საწყისი გამოსახულების გადაცემა სატელეფონო ხაზის საშუალებით, რომლის გამტარუნარიანობაა 14000 ბიტი/წმ. ამისათვის საჭირო იქნება 21000000 ბიტი/14000 ბიტი/წმ ≈ 3 წთ-ის ტოლი დრო. მაგრამ მონაცემთა 40-ჯერადი ($r=40$) კომპრესიისას კი ამისათვის საჭირო იქნება სულ 5 წმ!

მაგალითი 2. წყაროს შესაკუმშ მონაცემებად შევარჩიოთ გამოსახულების 4×4 ზომის ფრაგმენტი, რომელიც შეიცავს 4 ფერს: R ="წითელი", O ="ნარინჯისფერი", Y ="ლურჯი", G ="მწვანე" :

R	R	O	Y
R	O	O	Y
O	O	Y	G
Y	Y	Y	G

მოვახდინოთ ამ გამოსახულების სკანირება სტრიქონებად და თითოეულ ფერს მივანიჭოთ შესაბამისი ინტენსივობა, მაგალითად, $R = 3, O = 2, Y = 1$ u $G = 0$, რის შედეგადაც მივიღებთ მონაცემთა $X = (3,3,2,1,3,2,2,1,2,2,1,0,1,1,1,0)$ ვექტორს. მონაცემთა კომპესიისათვის გამოვიყენოთ კოდერი, რომლის მუშაობა ემყარება წყაროს მონაცემების კოდურ სიტყვებში გადაკოდირების შემდეგ ცხრილს (ცხრილის შერჩევის საკითხი გადავლოთ სამომავლოდ).

<i>კოდერი</i>	
ანათვლები	კოდური სიტყვა
3	001
2	01
1	1
0	000

კოდირების მოცემული ცხრილის გამოყენებით X ვექტორის თითოეული ელემენტი შევცვალოთ შესაბამისი კოდური მიმდევრობით (ეგრეთ წოდებული *კოდირება მახსოვრობის გარეშე*). კომპრესირებულ მონაცემებს (კოდურ $B(X)$ სიტყვას) ექნება შემდეგი სახე:

$$B(X) = (0,0,1,0,0,1,0,1,1,0,0,1,0,1,0,1,1,0,0,0,1,1,1,0,0,0).$$

ამასთანავე კომპესიის კოეფიციენტი $r=32/31 \approx 1,03$. შესაბამისად კომპრესიის სიჩქარე $R = 31/16 \approx 1,94$ ბიტი/ანათვალზე.

მაგალითი 3. ერთმანეთს შევადაროთ ორი სხვადასხვა კოდერი, რომლებიც ახორციელებენ მონაცემთა ერთი და იმავე ვექტორის შეკუმშვას (კომპრესიას).

$$X = ABRACADABRA.$$

პირველია კოდერი მახსოვრობის გარეშე, რომლის ანალოგიურიცაა განხილული წინა მაგალითში (X ვექტორის ყოველი ელემენტის კოდირება ხდება სხვა დანარჩენი ელემენტებისაგან დამოუკიდებლად – *კოდერი მახსოვრობის გარეშე*). მისთვის კოდირების ცხრილს ექნება შემდეგი სახე:

<i>კოდერი 1</i>	
სიმბოლო	კოდური სიტყვა
A	0
B	10
R	110
C	1110
D	1111

მეორე კოდური მიმდინარე სიმბოლოს კოდირებისას ითვალისწინებს მისი წინამდებარე (წინმსწრები) სიმბოლოს მნიშვნელობას, რის გამოც მიმდინარე **A** სიმბოლოსათვის კოდური სიტყვა სხვადასხვა **RA, DA** და **CA** შესაძლებლობებში იქნება სხვადასხვა, ანუ, სხვა სიტყვებით რომ ვთქვათ კოდს *გააჩნია მახსოვრობა* წყაროს ერთ სიმბოლოზე.

<i>კოდური 2</i>	
სიმბოლო, წინმსწრები სიმბოლო	კოდური სიტყვა
(A,-)	1
(B,A)	0
(C,A)	10
(D,A)	11
(A,R)	1
(R,B)	1
(A,C)	1
(A,B)	1

მონაცემთა **X = ABRACADABRA** ვექტორის შესაბამის კოდურ სიტყვებს ამ ორი ცხრილის გამოყენებით კოდირებისას ექნება შემდეგი სახე:

$$B_1(X) = 01011001110011110101100,$$

$$B_2(X) = 10111011111011.$$

ამრიგად, *კოდური 1*-ის (მახსოვრობის გარეშე) გამოყენებისას კომპრესიის სიჩქარეა $23/11 = 2,09$ ბიტი სიმბოლოზე მაშინ, როდესაც *კოდური 2*-სათვის იგი ტოლია $13/11=1,18$ ბიტისა სიმბოლოზე. ამის გამო მეორე კოდურის გამოყენება, მიუხედავად მისი სირთულისა, უფრო მიმზიდველია (მიზანშეწონილია).

2.2. კოდეზი მახსოვრობის ბარეში. ხაზმინის კოდეზი

უმარტივეს კოდეზს, რომელთა საფუძველზე შესაძლებელია მონაცემთა კომპრესიის განმარტოციელება, წარმოადგენენ კოდეზი მახსოვრობის ბარეში. ასეთ კოდეზში მონაცემთა საკოდირებელი ვექტორის ყოველი სიმბოლო იცვლება კოდეზი სიტყვით, რომელიც ეკუთვნის ორობითი სიმბოლოების ან სიტყვების პრეფიქსულ სიმრავლეს.

ორობითი S მიმდევრობების პრეფიქსული სიმრავლე ეწოდება ორობითი მიმდევრობების ისეთ სასრულო სიმრავლეს, რომელშიც ამ სიმრავლის არცერთი მიმდევრობა არ წარმოადგენს S -ის არცერთი სხვა მიმდევრობის პრეფიქსს, ანუ საწყისს.

მაგალითად, ორობითი სიტყვების $S1 = \{00, 01, 100, 110, 1010, 1011\}$ სიმრავლე წარმოადგენს ორობითი მიმდევრობების პრეფიქსულ სიმრავლეს, ვინაიდან თუ შევამოწმებთ S -ის 30 შესაძლო ერთობლივი ($w_i w_j$) კომბინაციიდან ნებისმიერ მათგანს, მაშინ ცხადი გახდება, რომ w_i არასოდეს იქნება w_j -ის პრეფიქსი (საწყისი). მეორეს მხრივ, $S2 = \{00, 001, 1110\}$ სიმრავლე არ წარმოადგენს ორობითი მიმდევრობების პრეფიქსულ სიმრავლეს, ვინაიდან 00 მიმდევრობა ამავე სიმრავლის სხვა (001) მიმდევრობის პრეფიქსია (საწყისია).

ამრიგად, თუ აუცილებელია მონაცემთა k ზომის A ალფაბეტიანი რომელიმე $X = (x_1, x_2, \dots, x_n)$ ვექტორის კოდირება, მაშინ კოდირება კოდით მახსოვრობის ბარეში ხორციელდება შემდეგნაირად:

– ადგენენ A ალფაბეტის $a_1, a_2, a_j \dots, a_k$ სიმბოლოების სრულ სიას, რომელშიც a_j არის X -ში სისშირის მიხედვით გამოჩენის j -რი სიმბოლო. ეს იმას ნიშნავს, რომ სიაში პირველი იქნება ალფაბეტში შედარებით ხშირად არსებული სიმბოლო, მეორე კი უფრო იშვიათად არსებული სიმბოლო და ა.შ;

– ყოველ a_j სიმბოლოს ორობითი S მიმდევრობების პრეფიქსული სიმრავლიდან უნიშნავენ (შეუსაბამებენ) კოდეზ w_j სიტყვას;

– კოდეზის გამოსასვლელზე ღებულობენ ყველა ორობითი სიტყვის გაერთიანების შედეგად ფორმირებულ საერთო მიმდევრობას.

პრეფიქსული სიმრავლეების ფორმირება და მათთან მუშაობა სიმრავლეთა თეორიის ცალკე სერიოზული თემაა, რომელიც აღნიშნული კურსის

ფარგლებს სცილდება, თუმცა მაინც მოგვიხდება რამდენიმე აუცილებელი შენიშვნის გაკეთება.

თუ $S = \{w_1, w_2, \dots, w_k\}$ პრეფიქსული სიმრავლეა, მაშინ შეიძლება განისაზღვროს ისეთი რიცხვებისაგან შედგენილი რაიმე $v(S) = (L_1, L_2, \dots, L_k)$, ვექტორი, რომლებიც წარმოადგენენ შესაბამისი პრეფიქსული მიმდევრობების სიგრძეებს (L_i - w_i -ის სიგრძეა).

(L_1, L_2, \dots, L_k) ვექტორს, რომელიც შედგება არაკლებადი დადებითი რიცხვებისაგან, უწოდებენ კრაფტის ვექტორს.

$$2^{-L_1} + 2^{-L_2} + \dots + 2^{-L_k} \leq 1. \quad (2.5)$$

ამ უტოლობას კრაფტის უტოლობა ეწოდება. მისთვის სამართლიანია შემდეგი მტკიცებულება: თუ S რომელიმე პრეფიქსული სიმრავლეა, მაშინ $v(S)$ კრაფტის ვექტორია. სხვა სიტყვებით რომ ვთქვათ, *პრეფიქსულ სიმრავლეში ორობითი მიმდევრობების სიგრძეები აკმაყოფილებენ კრაფტის უტოლობას.*

თუ (2.5) უტოლობა გადადის მტკიცე ტოლობაში, მაშინ ასეთ კოდს კომპაქტურს უწოდებენ. ამ კოდს გააჩნია მოცემული ალფაბეტის კოდებიდან უმცირესი სიგრძე, ანუ იგი ოპტიმალურია.

ქვემოთ მოყვანილია უმარტივესი პრეფიქსული სიმრავლეების მაგალითები და მათი შესაბამისი კრაფტის ვექტორები:

- $S1 = \{0, 10, 11\}$ და $v(S1) = (1, 2, 2)$;
- $S2 = \{0, 10, 110, 111\}$ და $v(S2) = (1, 2, 3, 3)$;
- $S3 = \{0, 10, 110, 1110, 1111\}$ და $v(S3) = (1, 2, 3, 4, 4)$;
- $S4 = \{0, 10, 1100, 1101, 1110, 1111\}$ და $v(S4) = (1, 2, 4, 4, 4, 4)$;
- $S5 = \{0, 10, 1100, 1101, 1110, 11110, 11111\}$ და $v(S5) = (1, 2, 4, 4, 4, 5, 5)$;
- $S6 = \{0, 10, 1100, 1101, 1110, 11110, 111110, 111111\}$ და $v(S6) = (1, 2, 4, 4, 4, 5, 6, 6)$.

დავუშვათ, რომ დასამუშავებელია კოდი მახსოვრობის გარეშე მონაცემთა იმ $X = (x_1, x_2, \dots, x_n)$ ვექტორის კომპრესიისათვის, რომლის k სიმბოლოს ზომის (k რაოდენობის სიმბოლოს შემცველი) ალფაბეტია A . განხილვისათვის შემოვიტანოთ ეგრეთ წოდებული სისშირეთა $F = (F_1, F_2, \dots, F_k)$ ვექტორი, სადაც F_i არის X ვექტორში A ალფაბეტის შედარებით ხშირად არსებული i -რი სიმბოლოს გამოჩენის რაოდენობაა. X -ის კოდირება მოვახდინოთ კოდით მახსოვრობის

გარეშე, რომლისთვისაც კრაფტის ვექტორია $L = (L_1, L_2, \dots, L_k)$. მაშინ კოდერის გამოსასვლელზე მიღებული ორობითი კოდური $B(X)$ მიმდევრობის სიგრძეა

$$L * F = L_1 * F_1 + L_2 * F_2 + \dots + L_k * F_k. \quad (2.6)$$

საუკეთესო კოდი მახსოვრობის გარეშე იქნებოდა კოდი, რომლისთვისაც $B(X)$ სიგრძე აღმოჩნდებოდა მინიმალური. ასეთი კოდის დამუშავებისთვის საჭიროა კრაფტის L ვექტორის პოვნა, რომლისთვისაც $L * F$ ნამრავლი იქნებოდა მინიმალური.

შესაძლო ვარიანტების მარტივი გადარჩევა კრაფტის ვექტორის პოვნისათვის საუკეთესო ხერხი არაა, განსაკუთრებით კი k -ს დიდი მნიშვნელობისათვის.

*ხაფმენის ალგორითმი, რომელიც მას ეწოდა მისი გამომგონებლის დევიდ ხაფმენის პატივსაცემად, იძლევა F -სათვის კრაფტის ოპტიმალური L ვექტორის პოვნის საშუალებას, ანუ ისეთი ვექტორისა, რომლისთვისაც $L * F$ ნამრავლი მინიმალურია. F -სათვის ოპტიმალური L -ის გამოყენებით მიღებულ კოდს ხაფმენის კოდს უწოდებენ.*

2.2.1. ხაფმენის ალგორითმი

ხაფმენის ალგორითმი რეალიზაციას უკეთებს სტატისტიკური კოდირების იდეას პრეფიქსული სიმრავლეების გამოყენებით და იგი მუშაობს შემდეგნაირად:

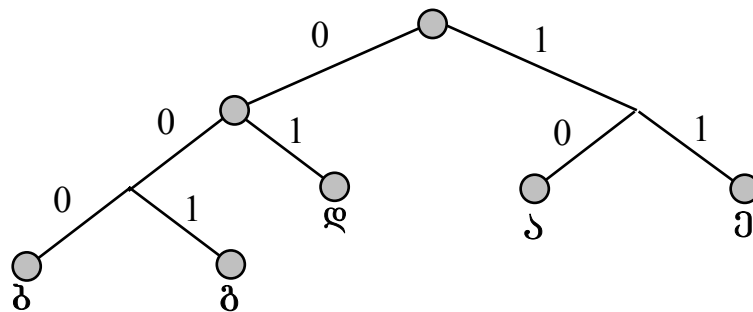
1. ალფავიტის ყველა სიმბოლო ამოვწეროთ ტექსტში მათი გამოჩენის ზრდადობის ან კლებადობის ალბათობის რიგითობის მიხედვით.

2. უმცირესი ალბათობების მქონე ორი სიმბოლო მიმდევრობით გაგაერთიანოთ ახალ შედგენილ სიმბოლოდ, რომლის გამოჩენის ალბათობა ჩავთვალოთ მისი შემდგენი სიმბოლოების ალბათობათა ჯამის ტოლად. საბოლოოდ ასეთნაირად აიგება ხე, რომლის თითოეულ კვანძს გააჩნია მის ქვემოთ არსებული ყველა კვანძის ჯამური ალბათობა.

3. მივსდით ხის ყოველი ფოთლისაკენ მიმავალ გზას ყოველი კვანძისაკენ მიმართულების მონიშვნით (მაგალითად, მარჯვნივ -1 , მარცხნივ -0). ასეთნაირად მიღებული მიმდევრობა ქმნის თითოეული სიმბოლოს შესაბამის კოდურ ხეს (ნახ. 2.4).

ავაგოთ კოდური ხე შემდეგი ალფაბეტის მქონე შეტყობინებისათვის:

ა	ბ	გ	დ	ე
10	5	8	13	10
ბ	გ	ა	ე	დ
5	8	10	10	13
ა	ე	ბგ	დ	
10	10	13	13	
ბგ	დ	აე		
13	13	20		
აე	ბგდ			
20	26			
აბბგ				
46				



ნახ. 2.4. ხაფმენის ალგორითმის მაილუსტრირებელი კოდური ხე

2.2.2. ენტროპიის საზღვრები ხაფმენის კოდისათვის

როგორც ცნობილია, არახელშემდგომიანი არხისათვის ფორმულირებული შენონის თეორემა X ვექტორის ენტროპიის საშუალებით განსაზღვრავს კომპრესიის (შეკუმშვის) ხარისხის ზედა საზღვარს. ხაფმენის კოდისათვის ვიპოვოთ კომპრესიის ხარისხის ზედა და ქვედა საზღვრები.

ვთქვათ, ხაფმენის კოდით მონაცემთა X ვექტორის კოდირებისას $R(X)$ კომპრესიის სიჩქარეა, რომელიც განისაზღვრება როგორც ფარდობა

$$R = k/n, \tag{2.7}$$

და რომელიც იზომება ერთ ასონიშანზე გადაანგარიშებულ ბიტებში (ორობითი სიმბოლოების რაოდენობაში). საჭიროა $R(X)$ -ის საზღვრების პოვნა, რომლებიც გამოსახულია X ვექტორის ენტროპიით.

ვთქვათ n სიგრძის მონაცემთა ვექტორია X და X -ში სიმბოლოების სიხშირეთა ვექტორია (F_1, F_2, \dots, F_k) . მაშინ ენტროპია

$$H(X) = \frac{1}{n} \sum_{i=1}^k F_i \log_2 \left(\frac{n}{F_i} \right). \quad (2.8)$$

შევეცადოთ $H(X)$ ენტროპიისა და კომპრესიის $R(X)$ ხარისხის (სიჩქარის) ერთმანეთთან შემდეგი სახით დაკავშირებას:

$$R(X) = \frac{1}{n} \sum_{i=1}^k F_i L_i, \quad (2.9)$$

სადაც (L_1, L_2, \dots, L_k) -ხაფშენის კოდის კრაფტის ზუსტი ვექტორია X -სათვის.

$y = \log_e X$ მრუდი ამობურცულია, ანუ ის ყოველთვის თავსდება საკუთარი მხების ქვეშ გარდა შეხების წერტილისა. კერძოდ, იგი არსებობს $y = x - 1$ წრფის ქვემოთ, გარდა $x=1$ წერტილისა, ვინაიდან ეს წრფე $y = \log_e X$ მრუდის მხებია წერტილში $x=1$. ამრიგად, შეიძლება ჩავწეროთ უტოლობა

$$\log_e X \leq x - 1, \quad x > 0, \quad x \neq 1. \quad (2.10)$$

(2.10) ფორმულაში ჩავსვათ $x = n \cdot 2^{-L_i} / F_i$ და ორივე მხარე გაგამრავლოთ F_i/n -ზე, რის შედეგადაც მივიღებთ

$$(F_i/n) \log_e \left(\frac{2^{-L_i}}{F_i/n} \right) \leq 2^{-L_i} - F_i/n. \quad (2.11)$$

უტოლობის ორივე მხარე (ნაწილი) ავჯამოთ i -ს მიხედვით. მარჯვენა ნაწილის ჯამი იქნება ნულის ტოლი, რადგანაც 2^{-L_i} და F_i/n ჯამებიდან თითოეული მათგანი ერთის ტოლია. შედეგად,

$$-\sum_{i=1}^k (F_i/n) \log_e (F_i/n) + \sum_{i=1}^k (F_i/n) \log_e (2^{-L_i}) \leq 0. \quad (2.12)$$

და ბოლოს, თუ ლოგარითმს e -ს ფუძით გადავიყვანოთ 2-ის ფუძიან ლოგარითმზე, მივიღებთ

$$H(X) \leq R(X). \quad (2.13)$$

განვიხილოთ კრაფტის $(L_1^*, L_2^*, \dots, L_k^*)$ ვექტორი, რომელშიც კოდური სიტყვების L_i^* სიგრძეები სიმბოლოთა სიხშირეებთან დაკავშირებულია შემდეგნაირად:

$$L_i^* \triangleq [-\log_2 (F_i / n)] . \quad (2.14)$$

მაშინ

$$R(X) = n^{-1} \sum_i F_i L_i \leq n^{-1} \sum_i F_i L_i^* \leq n^{-1} \sum_i F_i [-\log_2 (F_i / n) + 1] , \quad (2.15)$$

სადაც გათვალისწინებულია, რომ $[u] \leq u + 1$. უტოლობის მარჯვენა ნაწილი კი უდრის $(H(X) + 1)$ -ს.

ბოლოს კი ყოველივე ზემოთ მიღებულის გაერთიანებით მივიღებთ

$$H(X) \leq R(X) \leq H(X) + 1. \quad (2.16)$$

ხაფმენის მეთოდის ნაკლოვანებები

როგორც წინა მსჯელობიდან ჩანს, ხაფმენის კოდების შემთხვევაში ყველაზე დიდ სირთულეს წარმოადგენს შესაკუმში მონაცემების ყოველი ტიპისათვის ალბათობათა შესაბამისი ცხრილის არსებობის აუცილებლობა. ეს პრობლემას არ წარმოადგენს, თუ ცნობილია, რომ იკუმშება ქართული ან ინგლისური ტექსტი, როდესაც კოდერსა და დეკოდერს შეიძლება ქართული და ინგლისური ტექსტების შესაბამისი კოდური ხე. ზოგადად კი, როდესაც უცნობია შემავალი მონაცემების სიმბოლოთა ალბათობები, მაშინ ხაფმენის სტატისტიკური კოდები არაეფექტურია.

ამ პრობლემის გადაწყვეტას წარმოადგენს საკოდირებელი მონაცემების სტატისტიკური ანალიზი, რომელიც სრულდება მონაცემთა პირველი განხილვის საფუძველზე (პირველი ეტაპი) კოდური ხის აგებით. რაც შეეხება უშუალო კოდირებას, იგი ხდება მეორე ეტაპზე.

თუმცა, სიმართლეა, რომ არსებობს ხაფმენის კომპრესიის დინამიური ვერსია, რომელიც ხაფმენის კოდურ ხეს აგებს წაკითხვისა და აქტიური შეკუმშვის დროს. შესაბამისი ხე სისტემატურად ახლდება იმისათვის, რომ მან ასახოს შემავალი მონაცემების ალბათობათა ცვლილება. თუმცა პრაქტიკულად ეს ალგორითმიც გამოირჩევა სერიოზული შეზღუდულობით და, გარდა ამისა, უზრუნველყოფს კომპრესიის შედარებით მცირე ეფექტურობას.

ხაფმენის კოდის კიდევ ერთი ნაკლოვანებაა ის, რომ მათთვის კოდური სიტყვის მინიმალური სიგრძე შეუძლებელია ერთზე ნაკლები იყოს მაშინ, როდესაც სრულიად შესაძლებელია, რომ შეტყობინების ენტროპია ტოლი იყოს 0,1 და 0,01 ბიტის/ასონიშანზე. ამ შემთხვევაში ხაფმენის კოდი ხდება

მნიშვნელოვნად ჭარბი. პრობლემა შეიძლება გადაწყდეს ალგორითმის გამოყენებით სიმბოლოთა ბლოკებისადმი, მაგრამ ამ შემთხვევაში რთულდება კოდირება/დეკოდირების პროცედურა და მნიშვნელოვნად ფართოვდება კოდური ხე, რომელიც საბოლოო ჯამში უნდა შევინახოთ კოდთან ერთად.

და ბოლოს, ხაფმენის კოდი უზრუნველყოფს კოდის საშუალო სიგრძეს, რომელიც ენტროპიას ემთხვევა მხოლოდ იმ შემთხვევაში, როდესაც წყაროს სიმბოლოთა ალბათობები წარმოადგენენ ორის მთელ უარყოფით ხარისხებს: $1/2 = 0,5$; $1/4 = 0,25$; $1/8 = 0,125$; $1/16 = 0,0625$ და ა.შ. პრაქტიკაში კი ასეთი სიტუაცია საკმაოდ იშვიათად გვხვდება ან კიდევ იგი შეიძლება შეიქმნას სიმბოლოთა ბლოკირებით აქედან გამომდინარე ყველა შედეგით.

2.3. კოდები მახსოვრობით

ჩვეულებრივ განიხილავენ ორი ტიპის კოდებს მახსოვრობით:

- ბლოკური კოდები;
- კოდები სასრულო მახსოვრობით.

ბლოკური კოდები მონაცემთა ვექტორს ყოფს მოცემული სიგრძის ბლოკებად და შემდეგ თითოეულ ბლოკს ცვლიან ორობითი სიტყვების პრეფიქსული სიმრავლის კოდური სიტყვით. კოდური სიტყვების მიღებულ მიმდევრობას კოდერის გამოსასვლელზე აერთიანებენ ორობით სტრიქონში. ბლოკური კოდის შესახებ ამბობენ, რომ ის k -რი რიგის ბლოკური კოდია, თუ ყველა ბლოკის სიგრძეა k .

მაგალითისათვის მეორე რიგის ბლოკური კოდის გამოყენებით შევკუმშოთ მონაცემთა $X = (0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1)$ ვექტორი.

თავდაპირველად X ვექტორი დავეყოთ ბლოკებად, რომელთა სიგრძეებია 2: 01, 10, 10, 01, 11, 10, 01, 01.

ეს ბლოკები განვიხილოთ როგორც ახალი $\{01, 10, 11\}$ "ჰიპერალფაბეტის" ელემენტები. ახალი ალფაბეტის სიმბოლოებისადმი კოდის შესაბამისობის უზრუნველყოფის მიზნით განვსაზღვროთ ბლოკების მიმდევრობაში დამატებითი ალფაბეტის გამოჩენის სისშირეთა ვექტორი. ეს ვექტორია $(4, 3, 1)$, რაც იმას ნიშნავს, რომ ყველაზე უფრო ხშირად არსებული

ბლოკი 01 გამოჩნდა ოთხჯერ, გამოჩენის სიხშირის მიხედვით შემდეგი ბლოკი 10 – სამჯერ, ხოლო ყველაზე იშვიათი ბლოკი 11 – მხოლოდ ერთჯერ.

სიხშირეთა (4, 3, 1) ვექტორისათვის კრაფტის ოპტიმალური ვექტორია (1, 2, 2). ამრიგად, მონაცემთა განხილული X ვექტორის შემთხვევაში მეორე რიგის ოპტიმალური ბლოკური კოდისათვის კოდური განისაზღვრება 2.5 ცხრილის მიხედვით.

ცხრილი 2.5

კოდურის ცხრილი	
ბლოკი	კოდური სიტყვა
01	0
10	10
11	11

ყოველი ბლოკის შეცვლით მისი შესაბამისი კოდური სიტყვით მივიღებთ კოდური სიტყვების მიმდევრობას:

0, 10, 10, 0, 11, 10, 0, 0.

მათი გაერთიანებით კი მივიღებთ კოდურის გამომავალ მიმდევრობას:

$B(X) = (0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0)$.

შეიძლება შემოწმდეს, რომ საწყისი X ვექტორის $H(X)$ ენტროპია ტოლია **0.9887** ბიტისა/ასონიშანზე, ხოლო ბლოკური კოდის გამოყენების შედეგად მიღებული კომპრესიის (შეკუმშვის) ხარისხი $R(X) = 12/16 = 0.75$ ბიტს მონაცემთა ერთ ანათვალზე და იგი აღმოჩნდა ენტროპიის ქვედა საზღვარზე ნაკლები. მიღებული შედეგი თითქოსდა ეწინააღმდეგება შენონის თეორემას, რომელიც ამტკიცებს, რომ შეუძლებელია კოდის ისეთი საშუალო სიგრძის მიღწევა, რომელიც ნაკლებია წყაროს ენტროპიაზე. თუმცა სინამდვილეში ეს ასე არაა. თუ ყურადღებით შევხედავთ მონაცემთა X ვექტორს, მაშინ შევნიშნავთ შემდეგ კანონზომიერებას: სიმბოლო "0"-ის შემდეგ შედარებით ხშირია სიმბოლო "1"-ის არსებობა, ანუ პირობითი ალბათობა $P(1/0)$ მნიშვნელოვნად აღემატება უპირობო ალბათობას $P(0)$. შედეგად, ამ წყაროს ენტროპია უნდა ჩაითვალოს როგორც რთული შეტყობინების ენტროპია და იგი, როგორც ცნობილია, ყოველთვის ნაკლებია მარტივი შეტყობინებების ენტროპიასთან შედარებით.

მონაცემთა (X_1, X_2, \dots, X_n) ვექტორის კოდირების დროს კოდი სასრულო მახსოვრობით იყენებს კოდურ წიგნს, რომელიც შედგება რამდენიმე სხვადასხვა

კოდებისაგან მახსოვრობის გარეშე. მონაცემთა ყოველი ანათვალის კოდირება ხდება კოდური წიგნის იმ კოდით მახსოვრობის გარეშე, რომელიც განისაზღვრება მონაცემთა წინმსწრები (წინამდებარე) ანათვლების გარკვეული რაოდენობის სიდიდით.

მახსოვრობიანი კოდირების მაგალითად განვიხილოთ ადრე განხილული კლასიკური მაგალითი $X=ABRACADABRA$. სრულიად ნათელია, რომ ამ მიმდევრობაში ძლიერია სტატისტიკური კავშირი მის მიმდევრო სიმბოლოებს შორის, რომელიც აუცილებლად გასათვალისწინებელია კოდირების მეთოდის შერჩევისას.

კოდური მახსოვრობით მიმდინარე სიმბოლოს კოდირებისას ითვალისწინებს მისი წინმსწრები სიმბოლოს მნიშვნელობას. ამრიგად, კოდური სიტყვა მიმდინარე A სიმბოლოსათვის იქნება სხვადასხვა RA, DA და CA შესაძლებლობებში (სხვა სიტყვებით რომ ვთქვათ, კოდს გააჩნია მახსოვრობა წყაროს ერთ სიმბოლოზე) (ცხრილი 2.6).

ცხრილი 2.6

<i>კოდური</i>	
სიმბოლო, წინმსწრები სიმბოლო	კოდური სიტყვა
(A,-)	1
(B,A)	0
(C,A)	10
(D,A)	11
(A,R)	1
(R,B)	1
(A,C)	1
(A,B)	1

კოდირების შედეგია $B(X)=(10111011111011)$ ვექტორი, რომლის სიგრძეა 11 ბიტი და კომპრესიის სიჩქარეა $R = 13/11=1,18$ ბიტი მონაცემთა ერთ სიმბოლოზე მაშინ, როდესაც თანაბარი სამთანრიგა კოდით კოდირებისას $R=3$ -ის დროს საჭირო იქნებოდა 33 ბიტი.

2.4. არითმეტიკული კოდირება

ზემოთ განხილული მეთოდებისაგან განსხვავებით, არითმეტიკული კოდირებისას საკოდირებელი სიმბოლო (ან სიმბოლოთა ჯგუფი) იცვლება მათი

შესაბამისი კოდით, მთელი შეტყობინების კოდირების შედეგი იცვლება 0-დან 1-დე ინტერვალში ერთი რიცხვით ან ასეთი რიცხვების წყვილით. საწყისი ტექსტის კოდირების მცირდება მისი ამსახველი ინტერვალი, ხოლო მათი წარმომდგენი ათობითი (ან ორობითი) თანრიგების რაოდენობა იზრდება. შემავალი ტექსტის მორიგი სიმბოლოები ამცირებენ ინტერვალის სიდიდეს მოდულით განსაზღვრული მათი ალბათობების მნიშვნელობების მიხედვით. უფრო ალბათური სიმბოლოები ამას აკეთებენ უფრო ნაკლებად, ვიდრე ნაკლებ ალბათური სიმბოლოები და, ამის გამო, შედეგს ნაკლები რაოდენობის თანრიგებს უმატებენ.

ართმეტიკული კოდირების იდეა ნათელფყოთ მარტივი მაგალითის დახმარებით. ვთქვათ, რომ საჭიროა შემდეგი ტექსტური სტრიქონის კოდირება:

რადიოვიზირ.

კოდერის მუშაობის დაწყებამდე საკოდირებელი ტექსტის საწყისი ინტერვალია [0; 1).

საკოდირებელი შეტყობინების ალფაბეტი შეიცავს შემდეგ სიმბოლოებს (ასონიშნებს): {**ა, ბ, ვ, გ, დ, ე, ზ**}.

განვსაზღვროთ შეტყობინების ალფაბეტის თითოეული სიმბოლოს რაოდენობა (მათი ალბათობა) და თითოეულ მათგანს დაუნიშნოთ მათი ალბათობების შესაბამისი ინტერვალი. იმის გათვალისწინებით, რომ საკოდირებელ სიტყვაში სულ 10 ასონიშანია, შეიძლება შევადგინოთ 2.7 ცხრილი.

ცხრილი 2.7

სიმბოლო	ალბათობა	ინტერვალი
ა	0,1	0 – 0,1
ბ	0,1	0,1 – 0,2
ვ	0,1	0,2 – 0,3
გ	0,3	0,4 – 0,7
დ	0,1	0,3 – 0,4
ე	0,1	0,7 – 0,8
ზ	0,2	0,8 – 1

ცხრილში სიმბოლოთა განლაგება შესაძლებელია ნებისმიერი მიმდევრობით, კერძოდ: ტექსტში მათი გამოჩენის მიმდევრობის, ალფაბეტის ან ალბათობათა ზრდადობის მიხედვით და სხვა, რაც სულაც არაა პრინციპიალური საკითხი. ამასთან კოდირების შედეგი იქნება სხვადასხვა, მაგრამ ეფექტი – ერთი და იგივე.

2.4.1. კოდირება

ამრიგად, კოდირების დაწყებამდე საწყისი ინტერვალია [0 – 1).

შეტყობინების პირველი "რ" სიმბოლოს ნახვისას კოდური საწყის ინტერვალს ამცირებს ახალ [0,8 – 1) ინტერვალამდე, რომელიც გამოიყოფა აღნიშნული სიმბოლოსათვის. ამრიგად, პირველი ასონიშნის კოდირების შემდეგ კოდირების შედეგი მოთავსებული იქნება რიცხვების [0,8 – 1) ინტერვალში.

კოდერში შემავალი შეტყობინების მორიგი სიმბოლოა ასონიშანი "ა". ეს ასონიშანი შეტყობინებაში რომ ყოფილიყო პირველი, მაშინ მას გამოეყოფოდა

[0 – 0.1) ინტერვალი, მაგრამ სინამდვილეში იგი მოსდევს "რ"-ს და ამიტომ მისი კოდირება ხდება პირველი ასონიშნისათვის უკვე გამოყოფილი ინტერვალის შიგნით მოთავსებული ახალი ქვეინტერვალით, რის გამოც პირველი ასონიშნის შესაბამისი ინტერვალი მცირდება [0.80 – 0.82) მნიშვნელობამდე. სხვა სიტყვებით რომ ვთქვათ, "ა" ასონიშნისათვის გამოყოფილი [0 – 0.1) ინტერვალი ახლა თავსდება პირველი სიმბოლოს ("რ") მიერ დაკავებულ ინტერვალში (ახალი ინტერვალის დასაწყისი და დასასრული განისაზღვრება წინა ინტერვალის დასაწყისისადმი ამ ინტერვალის სიგანისა და მიმდინარე სიმბოლოსათვის გამოყოფილი ინტერვალის ნამრავლის მიმატებით). შედეგად ამისა მივიღებთ ახალ სამუშაო ინტერვალს [0.80 – 0.82), ვინაიდან წინა ინტერვალის სიგანე იყო 0,2, რომლის მეათედი ნაწილია 0,02.

შემდეგ "დ" სიმბოლოს შეესაბამება მისთვის გამოყოფილი [0.1 – 0.2) ინტერვალი, რაც უკვე არსებულ მუშა [0.80 – 0.82) ინტერვალს ამცირებს [0.802 – 0.804) მნიშვნელობამდე.

კოდერის შესასვლელზე მიწოდებული მომდევნო სიმბოლოა ასონიშანი "ო", რომლისთვისაც გამოყოფილია ფიქსირებული [0,3 – 0,6) ინტერვალი. უკვე არსებული მუშა [0.802 – 0.804) ინტერვალისათვის მივიღებთ ახალ მუშა [0,8026 - 0,8032) ინტერვალს და ა.შ.

ზემოაღნიშნულის საფუძველზე მივიღებთ:

საწყისი ინტერვალი	[0,0	–	1,0)
ანალიზის შემდეგ	[0,8	–	1,0)
	[0,80	–	0,82)
	[0,802	–	0,804)
	[0,8026	–	0,8032)
	[0,80302	–	0,80308)
	[0,803032	–	0,803038)
	[0,8030338	–	0,8030356)
	[0.80303488	–	0.80303506)
	[0.803034934	–	0.803034988)
	[0.8030349772	–	0.8030349880)

კოდირების შედეგია **[0,8030349772 – 0,8030349880]** ინტერვალი. სინამდვილეში კი, ცალსახა დეკოდირებისათვის ახლა უკვე საკმარისია ინტერვალის მხოლოდ ქვედა ან ზედა საზღვრის ცოდნა, რაც იმას ნიშნავს, რომ კოდირების შედეგის როლში შეიძლება გამოყენებული იყოს საბოლოო ინტერვალის დასაწყისი – **0,8030349772**. უფრო ზუსტად კი საბოლოო ინტერვალის შიგნით მოქცეული ნებისმიერი რიცხვის გამოყენებაა შესაძლებელი საწყისი შეტყობინების დეკოდირებისათვის, მაგალითად, რიცხვი **0,80303498**, რომელიც მოთავსებულია აღნიშნულ ინტერვალში. ამასთანავე ამ რიცხვს გააჩნია ათობითი თანრიგების შედარებით მცირე რაოდენობა, ვიდრე ქვედა და ზედა საზღვრების შესაბამის რიცხვებს და, შედეგად იგი შეიძლება წარმოვადგინოთ ორობითი თანრიგების უფრო მცირე რაოდენობით.

ნათელია, რომ რაც უფრო განიერია საბოლოო ინტერვალი, მით უფრო მცირე რაოდენობის ათობითი (და, შესაბამისად, ორობითი) თანრიგებით შეიძლება ის წარმოვადგინოთ. ინტერვალის სიგანე კი დამოკიდებულია საკოდირებელი სიმბოლოების ალბათობათა განაწილებაზე – უფრო ალბათური სიმბოლოები ინტერვალს ნაკლებად ამცირებენ და, შესაბამისად, კოდირების

შედგეს უმატებენ ნაკლები რაოდენობის ბიტებს. ეს ვახვენოთ მარტივ მაგალითზე.

დავუშვათ, რომ საჭიროა სიმბოლოთა **ა ა ა ა ა ა ა ა ა** # სტრიქონის კოდირება, სადაც **"ა"** ასონიშნის ალბათობაა 0,9. ამ შემთხვევაში მოცემული სტრიქონის კოდირების პროცედურა და მიღებული შედეგი იქნება შემდეგი:

შემაჯავალი სიგნალი	ქვედა საზღვარი	ზედა საზღვარი
	0,0	1,0
ა	0,0	0,9
ა	0,0	0,81
ა	0,0	0,729
ა	0,0	0,6561
ა	0,0	0,59049
ა	0,0	0,531441
ა	0,0	0,4782969
ა	0,0	0,43046721
ა	0,0	0,387420489
#	0,3486784401	0,387420489

კოდირების შედეგი შეიძლება იყოს, მაგალითად, **0,35**, რომელიც მთლიანად ექცევა სასრულო **0.3486784401 – 0.387420489** ინტერვალის შიგნით. ამ რიცხვის ორობითი წარმოდგენისათვის საჭიროა & ბიტი (ორი ათობითი თანრიგი დაახლოებით შეესაბამება 7 ორობით თანრიგს), მაშინ, როდესაც კოდირების შედეგის ორობითი წარმოდგენისათვის წანა მაგალითისათვის (**0,80303498**) საჭიროა 27 ბიტი!!

2.4.2. დეკოდირება

დავუშვათ, რომ ყველაფერი ის, რაც ცნობილია დეკოდერისათვის ტექსტის შესახებ, ესაა სასრულო ინტერვალი **[0,8030349772–0,8030349880]**. დეკოდერისათვის, ისევე როგორც კოდერისათვის, ცნობილია აგრეთვე ალფაბეტისათვის გამოყოფილი ინტერვალების განაწილების ცხრილი. დეკოდერი მაშინვე არკვევს, რომ პირველი კოდირებული სიმბოლოა **"რ"**, ვინაიდან კოდირების შედეგი მთლიანად მოთავსებულია [0.8 – 1), რომელიც აღნიშნული სიმბოლოსათვისაა გამოყოფილი ცხრილში.

ახლა გავიმეოროთ კოდერის მოქმედებები:

დასაწყისში [0.0 – 1.0);

ანალიზის შემდეგ [0.8 – 1.0).

კოდირების შედეგიდან გამოვრიცხავთ უკვე ცნობილი პირველი "რ" სიმბოლოს გავლენას. ამისათვის კოდირების შედეგს გამოვაკლოთ "რ"-სათვის გამოყოფილი დიაპაზონის ქვედა საზღვარი – $(0,8030349772 - 0.8 = 0,0030349772)$ და მიღებული შედეგი გავყოთ "რ"-სათვის გამოყოფილი ინტერვალის სიგანეზე, ანუ $0,2$ -ზე. ამის შედეგად მივიღებთ $0,0030349772/0,2=0,015174886$. ეს რიცხვი მთლიანად თავსდება "ა" ასონიშნისთვის გამოყოფილ $[0 - 0,1)$ ინტერვალში. შესაბამისად, დეკოდირებული მიმდევრობის მეორე სიმბოლო იქნება "ა".

ვინაიდან ჩვენ უკვე ვიცით ორი დეკოდირებული ასო – რა, ამიტომ ამ ინტერვალიდან გამოვრიცხოთ "ა" ასონიშნის გავლენა. ამ მიზნით $0,015174886$ ნარჩენს გამოვაკლოთ "ა" ასონიშნის ინტერვალის შესაბამისი ქვედა საზღვარი, ანუ $0,015174886 - 0.0 = 0,015174886$ და ეს შედეგი გავყოთ "ა" ასონიშნისათვის გამოყოფილი ინტერვალის სიგანეზე, ანუ $0,1$ -ზე: $0,015174886/0,1=0,15174886$. მიღებული შედეგი მოთავსებულია "დ" ასონიშნისათვის გამოყოფილ დიაპაზონში, რის გამოც მორიგი დეკოდირებული ასონიშანი იქნება "დ".

კოდირების შედეგიდან გამოვრიცხოთ "დ" ასონიშნის გავლენა. მივიღებთ $(0,15174886 - 0,1)/0,1 = 0,5174886$. ეს შედეგი ხვდება "ო" ასონიშნისათვის გამოყოფილ ინტერვალში და, შედეგად, მორიგი დეკოდირებული სიმბოლო იქნება "ო" და ა.შ. მანამ, სანამ არ მოხდება ყველა სიმბოლოს დეკოდირება.

დეკოდირებული რიცხვი	სიმბოლო გამოსასვლელზე	საზღვრები ქვედა	რები ზედა	ინტერვალის სიგანე
0,8030349772	რ	0,8	1,0	0,2
0,015174886	ა	0,0	0,1	0,1
0,15174886	დ	0,1	0,2	0,1
0,5174886	ო	0,3	0,6	0,3
0,724962	ო	0,7	0,8	0,1
0,24962	ვ	0,2	0,2	0,1
0,4962	ო	0,3	0,6	0,3
0,654	ზ	0,6	0,7	0,1
0,54	ო	0,3	0,6	0,3
0,80	რ	0,6	0,8	0,2
0.0	დეკოდირების დასასრული			

ესაა არითმეტიკული კოდირების ძირითადი იდეა და მისი პრაქტიკული რეალიზაცია შედარებით რთულია. ზოგიერთი პრობლემა შეიძლება შევნიშნოთ

უშუალოდ მოყვანილი მაგალითიდან. პირველი მათგანი იმაში მდგომარეობს, რომ დეკოდერს აუცილებლად რაიმე ხერხით უნდა ვაცნობოთ დეკოდირების პროცესის დამთავრების შესახებ, ვინაიდან ნარჩენი 0,0 შეიძლება ნიშნავდეს ასონიშან "ა"-ს ან აა, ააა, აააა და ა.შ. მიმდევრობებს. ამ პრობლემის გადაწყვეტა შესაძლებელია ორი ხერხით: პირველი მათგანი მდგომარეობს იმაში, რომ მონაცემთა კოდის გარდა შესაძლოა საკოდირებელი მასივის ზომის განმსაზღვრელი რიცხვის შენახვა. ამ შემთხვევაში დეკოდირების პროცესი შეწყდება მაშინ, როდესაც მის გამოსასვლელზე მასივი მიაღწევს აღნიშნულ ზომას. მეორე ხერხი გულისხმობს წყაროს მოდელში ბლოკის დასრულების მიმანიშნებელი სპეციალური სიმბოლოს, მაგალითად, #-ს ჩასმას, რის გამოც დეკოდერის გამოსასვლელზე ამ სიმბოლოს გამოჩენისთანავე შეწყდება დეკოდირება.

მეორე პრობლემა გამომდინარეობს თვით არითმეტიკული კოდირების იდეიდან და ის იმაში მდგომარეობს, რომ კოდირების საბოლოო შედეგი – საბოლოო ინტერვალი – ცნობილი გახდება მხოლოდ კოდირების პროცესის დამთავრებისას. ამიტომ ისმის კითხვა – არ შეიძლება კოდირებული შეტყობინების გადაცემა მანამ, სანამ არ მოხდება საწყისი შეტყობინების უკანასკნელი ასონიშნის მიღება და არ იქნება განსაზღვრული საბოლოო ინტერვალი? სინამდვილეში ასეთი დაყოვნების აუცილებლობა არაა. კოდირების შედეგის ამსახველი ინტერვალის შემცირების კვალბაზე მისი უფროსი ათობითი ნიშნები (ან ორობით ფორმაში ჩაწერის შემთხვევაში მისი უფროსი ბიტები) წყვეტენ ცვლილებას (იხილეთ კოდირების მოყვანილი მაგალითი). ამის სედეგად ეს თანრიგები (ან ბიტები) უკვე შეიძლება გადაიცეს. ამრიგად, კოდირებული მიმდევრობის გადაცემა ხორციელდება მხოლოდ გარკვეული მცირე შეყოვნებით, რაც უმნიშვნელოა და არაა დამოკიდებული საკოდირებელი შეტყობინების ზომაზე.

და მესამე პრობლემა – ესაა ინტერვალის წარმოდგენის სიზუსტის საკითხი. მოყვანილი მაგალითიდან ჩანს, რომ ინტერვალის წარმოდგენის სიზუსტე (მისი წარმოდგენისათვის საჭირო ათობითი თანრიგების რაოდენობა) შეუზღუდავად იზრდება საკოდირებელი შეტყობინების სიგრძის გაზრდისას. ეს პრობლემა ჩვეულებრივ წყდება სასრულო თანრიგებიანი არითმეტიკის გამოყენებითა და რეგისტრების თანრიგების გადავსებისადმი კონტროლის განხორციელებით.

2.5. კოდირების ლექსიკონური მეთოდები. ზივა-ლემპელის მეთოდი

პრაქტიკულად ყველა ლექსიკონური მეთოდი ეკუთვნის ებრაელი მეცნიერების ზივასა და ლემპელის მიერ 1977 წელს გამოქვეყნებულ შრომაში აღწერილ ალგორითმებს. მათი არსი იმაში მდგომარეობს, რომ შესაკუმში ტექსტის ფრაზები იცვლება იმ ადგილის მაჩვენებლით, სადაც ისინი ადრე არსებობდა მოცემულ ტექსტში.

ალგორითმების ეს ოჯახი იწოდება ზივა-ლემპელის სახელით და აღინიშნება როგორც **LZ**-კომპრესია (შეკუმშვა). ეს მეთოდი სწრაფად ეწეობა ტექსტის სტრუქტურას და მას შეუძლია მოკლე ფუნქციონალური სიტყვების კოდირება, რომლებიც ხშირად ჩნდებიან ტექსტში. ახალი სიტყვები და ფრაზები შეიძლება ასევე ფორმირდეს ტექსტში ადრე არსებული ნაწილებისაგან.

კომპრესირებული (შეკუმშული) ტექსტის დეკოდირება ხორციელდება პირდაპირ – მიღებული მაჩვენებლის შეცვლით ლექსიკონის იმ მზა ფრაზით, რომელზედაც მიუთითებს მაჩვენებელი. პრაქტიკაში **LZ**-მეთოდით მიიღწევა კარგი კომპრესია და მის უმთავრეს თავისებურებას წარმოადგენს დეკოდერის ძალიან სწრაფი მუშაობა. (რდესაც ჩვენ ვსაუბრობთ ტექსტზე, ვთვლით, რომ ხდება მონაცემთა რომელიმე სასრულო ალფაბეტიანი ვექტორის კოდირება და სულაც არაა აუცილებელი, რომ ეს მაინც და მაინც იყოს ტექსტი).

კოდირების ლექსიკონური მეთოდების უმრავლესობა ატარებს იდეის ავტორების ზივასა და ლემპელის სახელს და ხშირად მიაჩნიათ, რომ ყველა მათგანი იყენებს კოდირების ერთსა და იმავე ალგორითმს. სინამდვილეში კი ამ ალგორითმების ოჯახის სხვადასხვა წარმომადგენელი ძალზე ძლიერ განსხვავდებიან ერთმანეთისაგან მუშაობის დეტალებში.

კოდირების ყველა ლექსიკონური მეთოდი შეიძლება დავეთოთ ორ ჯგუფად.

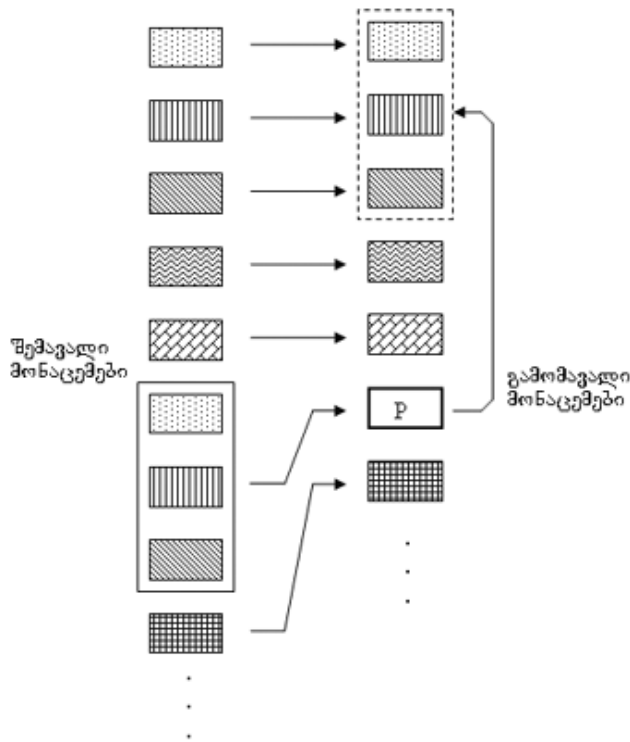
პირველი ჯგუფის მეთოდები, საკოდირებელ მიმდევრობაში აღმოაჩენენ რა სიმბოლოთა ადრე არსებულ ჯგუფს, იმის ნაცვლად, რომ გაიმეორონ ეს ჯგუფები, მათ ცვლიან წინა გამეორებების მაჩვენებლებით.

ალგორითმების ამ ჯგუფში ლექსიკონი არაცხადი სახით არსებობს დასამუშავებელ მონაცემებში, ამიტომ ინახება მხოლოდ გამეორებადი სიმბოლოების ჯგუფების მაჩვენებლები.

ამ ჯგუფის ყველა მეთოდი, როგორც უკვე იყო აღნიშნული, ემყარება აბრაჰამ ლემპელისა და იაკობ ზივას მიერ დამუშავებულ და 1977 წელს

გამოქვეყნებულ **LZ77** ალგორითმს. ამ ჯგუფის შედარებით სრულყოფილი წარმომადგენელია სტორერისა და შიმანსკის მიერ 1982 წელს გამოქვეყნებული ალგორითმი **LZSS**, რომელიც მოიცავს ამ მიმართულებით მიღებულ ყველა მიღწევას.

ამ ჯგუფის ალგორითმების კოდირების პროცედურა წარმოდგენილია 2.5 ნახაზზე.

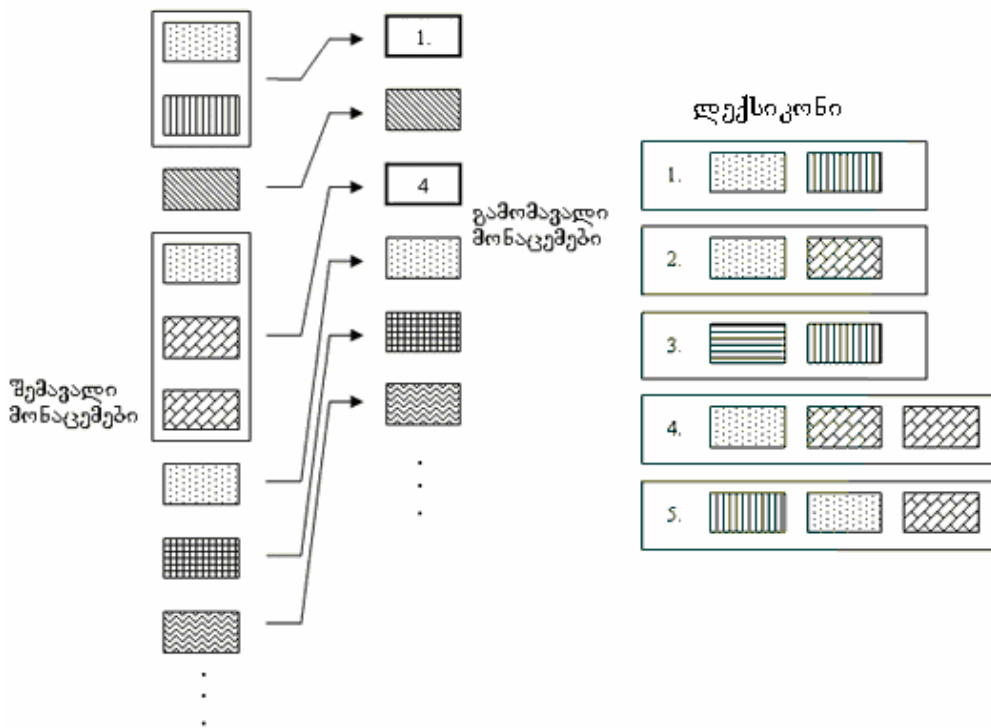


ნახ. 2.5. ლექსიკონური მეთოდის პირველი ჯგუფით კოდირების პროცედურა

მეორე ჯგუფის ალგორითმები წყაროს საწყის ლექსიკონის გარდა დამატებით ქმნიან იმ ფრაზების ლექსიკონს, რომლებიც წარმოადგენენ საწყისი ლექსიკონის სიმბოლოების გამეორებად კომბინაციებს, რომლებიც გვხვდება საწყის მონაცემებში. ამასთანავე ლექსიკონის ზომა იზრდება, რის გამოც მისი კოდირებისათვის საჭირო იქნება ბიტების უფრო მეტი რაოდენობა. მაგრამ ამ ლექსიკონის მნიშვნელოვანი ნაწილი უკვე იქნება არა ცალკეული ასონიშნები, არამედ – მათი შესამებები ან მთლიანი სიტყვები. იმ შემთხვევაში, როდესაც კოდერი აღმოაჩენს ადრე არსებულ ფრაზას, ის მას ცვლის ლექსიკონის ამ ფრაზის შემცველი ინდექსით. ამის გამო ინდექსის კოდის სიგრძე მიიღება უფრო მცირე ან გაცილებით მცირე ფრაზის კოდის სიგრძესთან შედარებით.

ამ ჯგუფის ყველა მეთოდი ემყარება ზივასა და ლემპელის მიერ დამუშავებულ და 1978 წელს გამოქვეყნებულ **LZ78** ალგორითმს. დღეისათვის ლექსიკონური მეთოდების ამ ჯგუფის შედარებით სრულყოფილი წარმომადგენელია ტერი ველჩემის მიერ 1984 წელს დამუშავებული **LZW**

ალგორითმი. აღნიშნული ჯგუფის ალგორითმების იდეა შეიძლება აიხსნას 2.6 ნახაზის მეშვეობით.



ნახ. 2.6. ლექსიკონური მეთოდის მეორე ჯგუფით კოდირების პროცედურა

მეორე ჯგუფის ალგორითმების მუშაობის ასხნა შედარებით მარტივია, რის გამოც **LZ**-კოდერების მოქმედების პრინციპის განხილვა დავიწყეთ **LZW** ალგორითმიდან. ზოგადად განვიხილოთ **LZW**-კოდერისა და დეკოდერის მუშაობა.

2.5.1. კოდირება

კოდირების პროცესი საკმაოდ მარტივია. მიმდევრობით ვკითხულობთ შემაჯავალი ნაკადის (სტრიქონის) სიმბოლოებს და ვამოწმებთ, არის თუ არა ჩვენს მიერ უკვე შექმნილ ცხრილში ასეთი სტრიქონი. თუ ასეთი სტრიქონი არის, მაშინ ვკითხულობთ მორიგ სიმბოლოს, ხოლო თუ არაა, მაშინ გამომავალ ნაკადში შეგვაქვს წინა ნაპოვნი სტრიქონის კოდი, ეს სტრიქონი შეგვაქვს ცხრილში და ხელახლა ვიწყებთ ძებნას.

ვთქვათ, კოდერის შესასვლელს მიეწოდება სიმბოლოთა / **WED** / **WE** / **WEE** / **WEB** სახის მიმდევრობა, ამასთანავე შემაჯავალი სიმბოლოების ალფაბეტის ზომა $dim A = 255$. კომპრესიის (შეკუმშვის) სქემას აქვს შემდეგი სახე:

შემავალი სიმბოლოები	გამომავალი კოდი	ლექსიკონის ახალი სიმბოლოები
<i>/W</i>	<i>/</i>	<i>256 = /W</i>
<i>E</i>	<i>W</i>	<i>257 = WE</i>
<i>D</i>	<i>E</i>	<i>258 = ED</i>
<i>/</i>	<i>D</i>	<i>259 = D/</i>
<i>WE</i>	<i>256</i>	<i>260 = /WE</i>
<i>/</i>	<i>E</i>	<i>261 = E/</i>
<i>WEE</i>	<i>260</i>	<i>262 = /WEE</i>
<i>/W</i>	<i>261</i>	<i>263 = E/W</i>
<i>EB</i>	<i>257</i>	<i>264 = WEB</i>
<i><END></i>	<i>B</i>	

შედგება მივიღებთ გამომავალ კოდს

$$/WED<256>E<260><261><257>B.$$

ისმება კითხვა, ამ შემთხვევაში როგორ შეიცვალა გამომავალი კოდის სიგრძე შემავალთან შედარებით?

15 ასონიშნიანი სიგრძისა და $\dim A = 255$ ზომის მქონე */WED / WE / WEE / WEB* სტრიქონის ორობითი კოდირებისათვის თუ დაგვჭირდებოდა $15 \cdot \log_2 255 = 15 \cdot 8 = 120$ ბიტი, მაშინ კოდერის 264 ასონიშნისაგან შედგენილი ალფაბეტის 10 ახალი სიმბოლოს შესაბამისი სიგრძის მქონე გამომავალი *WED <256> E <260> <261> <257> B* სტრიქონის ორობითი კოდირებისათვის საკმარისია $10 \cdot 9 = 90$ ბიტი.

LZW-კოდერი მუშაობას იწყებს *4K* ზომის ლექსიკონიდან, რომელიც მისამართების მიხედვით შეიცავს ცალკეული ასონიშნებისათვის 0-დან 255-დე და ქვესტრიქონებისათვის 256-დან 4095-დე მიმართვებს. კოდირების პროცესში ტექსტი ექვემდებარება ქვესტრიქონებად დაშლას, სადაც ყოველი ახალი ქვესტრიქონი ყველაზე გრძელია (უკვე განხილულ სტრიქონებს დამატებული ერთი სიმბოლო)-სთან შედარებით. იგი კოდირდება როგორც მის პრეფიქსს დამატებული დამატებითი სიმბოლო, რის შემდეგაც ახალი ქვესტრიქონი ემატება ლექსიკონს და მისაღმი შესაძლებელი იქნება მიმართვა მომავალში.

2.5.2. დეკოდირება

კოდირებული მონაცემების დამუშავების შედეგად *LZW*-დეკოდერი მისგან აღადგენს საწყის მონაცემებს. ისევე როგორც შეკუმშვის (კომპრესიის) ალგორითმი, დეკოდერი ახალ სტრიქონებს უმატებს ყოველთვის, როდესაც

შემავალ ნაკადში აღმოაჩენს ახალ კოდს. მან შემავალი კოდი უნდა გარდაქმნას სიმბოლოთა გამომავალ სტრიქონად და ეს უკანასკნელი გადასცეს კოდერის გამოსასვლელს.

LZW–დეკოდერის მუშაობის ალგორითმი:

სტრიქონი კოდერის გამოსასვლელზე – **/WED<256>E<260><261><257>B**.

შემავალი სიმბოლოები	გამომავალი სტრიქონი	ლექსიკონის ახალი სიმბოლოები
/	/	
W	W	256 = /W
E	E	257 = WE
D	D	258 = ED
256	/W	259 = D/
E	E	260 = /WE
260	/WE	261 = E/
261	E/	262 = /WEE
257	WE	263 = E/W
B	B	264 = WEB

კომპრესიის ამ მეთოდის ყველაზე მნიშვნელოვანი ღირსებაა ის, რომ ახალი სიმბოლოების მთელი ლექსიკონი დეკოდერს მიეწოდება საკუთრივ გადაცემის გარეშე. დეკოდირების პროცესის დასასრულს დეკოდერს გააჩნია ახალი სიმბოლოების ზუსტად ისეთივე ლექსიკონი, რომელიც კოდირების პროცესში დააგროვა კოდერმა, ამასთანავე მისი შექმნა იყო დეკოდირების პროცესის ნაწილი.

რამდენადმე განსხვავებულია **LZ**–მეთოდის პირველი გამოქვეყნებული ვერსიის – **LZ77**–ის ოჯახის კოდერი/დეკოდერის მუშაობა.

LZ77–ის ალგორითმში მაჩვენებლები მიაჩნებიან ფრაზებს მუდმივი ზომის იმ ფანჯარაში, რომელიც წინ უძღვის კოდის პოზიციებს. მაჩვენებლებით შეცვლილი ქვესტრიქონის მაქსიმალური სიგრძე განისაზღვრება **F** პარამეტრით (ჩვეულებრივ გი მოთავსებულია 10-დან 20-დე შუალედში). ეს შეზღუდვები საშუალებას გვაძლევს გამოვიყენოთ ეგრეთ წოდებული **N** სიმბოლოსაგან შედგენილი "მცოცავი ფანჯარა". მათგან პირველი **N-F** უკვე იყო კოდირებული, ხოლო ბოლო **F** სიმბოლოები ქმნიან წინმსწრებ ბუფერს.

პირველ $N-F$ სიმბოლოებში კოდირებისას ფანჯრები ეძებენ ბუფერის შესაბამის ყველაზე გრძელ სტრიქონს. მან ნაწილობრივ შეიძლება გადაფაროს (გადააფაროს) ბუფერი, მაგრამ თვითონ შეუძლებელია იყოს ბუფერი.

ყველაზე მეტად ნაპოვნი შესაბამისობა შემდეგ კოდირდება $[i, j, a]$ ტრიადით, სადაც i მისი წანაცვლებაა ბუფერის დასაწყისიდან, j – შესაბამისობის სიგრძეა, ხოლო a – პირველი სიმბოლოა, რომელიც არ შეესაბამება ფანჯრის ქვესტრიქონს. ამის შემდეგ ფანჯარა გაიწვევა მარჯვნივ $j+1$ სიმბოლოზე და იგი მზადაა ალგორითმის ახალი (მომდევნო) ბიჯისათვის (ეტაპისათვის).

ყოველი მაჩვენებლისადმი გარკვეული სიმბოლოს მიბმა (დაკავშირება) იძლევა იმის საშუალებას, რომ კოდირება ჩატარდება იმ შემთხვევაშიც კი, თუ შესაბამისობა არ იქნება ნაპოვნი წინმსწრები ბუფერის პირველი სიმბოლოსათვის.

კოდერისა და დეკოდერისათვის საჭირო მახსოვრობის მოცულობა (ტევა-დობა) იზღუდება ფანჯრის ზომით. ტრიადაში (i) წანაცვლების წარმოდგენისათვის საჭირო ბიტების რაოდენობაა $[\log(N-F)]$. ტრიადის მიერ შეცვლილი სიმბოლოების (j) რაოდენობა შეიძლება კოდირებული იყოს $[\log F]$ რაოდენობის ბიტით.

დეკოდირება ხორციელდება ძალზე სწრაფად და მარტივად. ამასთანავე სენარჩუნებულია ფანჯარასთან მუშაობის იგივე თანმიმდევრობა, რაც იყო კოდირებისას, მაგრამ ერთნაირი სტრიქონების ძებნისაგან განსხვავებით იგი, პირიქით, იგი ფანჯრიდან ახდენს მის კოპირებას მომდევნო ტრიადის შესაბამისად.

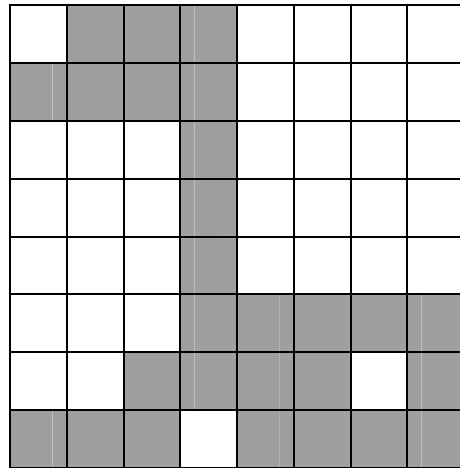
2.6. ბამეორებათა სიბრძის კოდირება

უბნების (ანუ გამეორებების) სიგრძეთა კოდირება შეიძლება საკმაოდ ეფექტური იყოს ორობითი მონაცემების (მაგალითად, შავ-თეთრი ფაქსიმილური გამოსახულებების, სწორი ხაზებისა და ერთგვაროვანი უბნების შემცველი შავ-თეთრი გამოსახულებების, სქემების და ა.შ.) კომპრესიისას (შეკუმშვისას). გამეორებათა სიგრძის კოდირება წარმოადგენს უძრავ გამოსახულებათა კომპრესიის ფართოდ გავრცელებული **JPEG** ალგორითმის ერთ-ერთ ელემენტს.

მონაცემთა კომპრესიის იდეა გამეორებათა სიგრძის კოდირების საფუძველზე იმაში მდგომარეობს, რომ საკუთრივ მონაცემთა კოდირების

ნაცვლად კოდირებას უკეთებენ იმ უბნების სიგრძეთა შესაბამის რიცხვებს, რომლებზეც მონაცემები ინარჩუნებენ უცვლელ მნიშვნელობას.

დავუშვათ, რომ საჭიროა **8x8** ზომის ორობითი (ორფერიანი) გამოსახულების კოდირება (ნახ. 2.7).



ნახ. 2.7. ორფეროვანი **8x8** ზომის გამოსახულება

მოვახდინოთ ამ გამოსახულების სტრიქონული სკანირება (გამოსახულებაზე ორ ფერს შეესაბამება 0 და 1), რის შედეგადაც მივიღებთ მონაცემთა ორობით ვექტორს

$$X=(0111000011110000000100000001000000010000000111100011110111101111),$$

რომლის სიგრძეა 64 ბიტი (საწყისი კოდის სიხარვე 1 ბიტი გამოსახულების ერთ ელემენტზე).

X ვექტორში გამოვყოთ უბნები, რომლებზეც მონაცემები უცვლელია, და განვსაზღვროთ მათი სიგრძეები. უბანთა სიგრძეების r მიმდევრობა წარმოვადგინოთ მთელი დადებითი რიცხვების სახით:

$$r=(1, 3, 4, 4, 7, 1, 7, 1, 7, 1, 7, 4, 3, 4, 1, 4, 1, 4) .$$

შემდეგ ეს მიმდევრობა, რომელშიც შეინიშნება გარკვეული გამეორადობა (ერთიანებისა და ოთხიანების რაოდენობა გაცილებით აღემატება სხვა სიმბოლოების რაოდენობას), შეიძლება კოდირებული იყოს რაიმე სტატისტიკური, მაგალითად, ხაფმენის კოდით მახსოვრობის გარეშე, რომლის შესაბამისი ცხრილია 2.8.

ცხრილი 2.8

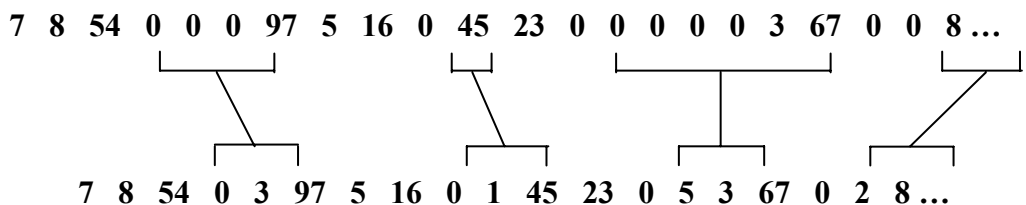
კოდური	
უბნის სიგრძე	კოდური სიტყვა
4	0
1	10
7	110
3	111

იმის საჩვენებლად, რომ საკოდირებელი მიმდევრობა იწყება ნულიდან, კოდური სიტყვის დასაწყისში დავამატოთ პრეფიქსული სიმბოლო **0**. ამის სდეგად მივიღებთ 34-ბიტის კოდურ სიტყვას

$$B(r) = (0100011010110101101011001110100100),$$

ანუ კოდის ჯამური სიჩქარე იქნება $R=34/64$, ანუ გამოსახულების ერთ ელემენტზე 0,5 ბიტზე ოდნავ მეტი. მრავალი განმეორებადი ელემენტების შემცველი უფრო დიდი ზომის გამოსახულებათა კომპრესიის შემთხვევაში შეკუმშვის (კომპრესიის) ეფექტურობა შეიძლება მნიშვნელოვნად უფრო მაღალი აღმოჩნდეს.

ქვემოთ მოყვანილია განმეორებათა სიგრძეების კოდირების გამოყენების მაგალითი იმ შემთხვევაში, როდესაც ციფრულ მონაცემებში გვხვდება ნულოვანი მნიშვნელობების დიდი რაოდენობა. ყოველთვის, როდესაც მონაცემთა ნაკადში გვხვდება "ნული", მისი კოდირება ხდება ორი რიცხვით. მათგან პირველია **0**, რომელიც წარმოადგენს ნულების ნაკადის სიგრძის კოდირების დასაწყისის მაჩვენებელს. თუ ჯგუფში ნულების საშუალო რაოდენობა ორზე მეტია, მაშინ განხორციელდება შეკუმშვა. მეორეს მხრივ, ცალკეული ნულების დიდმა რაოდენობამ შეიძლება მიგვიყვანოს საკოდირებელი ფაილის ზომის გაზრდამდეც კი:



გამოსახულებათა და ბგერითი სიგნალების შეკუმშვის (კომპრესიის) მიზნით არარღვევადი კოდირების ფართოდ გამოყენებად და მარტივ მეთოდს წარმოადგენს დიფერენციალური კოდირების მეთოდი.

2.7. დიფერენციალური კოდირება

დიფერენციალური კოდერის მუშაობა ეყრდნობა იმ ფაქტს, რომ მონაცემთა მრავალი ტიპისათვის მეზობელ ანათვლებს შორის სხვაობა შედარებით მცირეა საკუთრივ მონაცემთა დიდი მნიშვნელობის შემთხვევაშიც კი. მაგალითად, არ შეიძლება ველოდეთ დიდ სხვაობებს ციფრული გამოსახულების მეზობელ პიქსელებს (ელემენტებს) შორის.

შემდეგი მარტივი მაგალითი გვიჩვენებს დიფერენციალური (მეზობელ ანათვლებს შორის სხვაობის) კოდირების მეთოდის უპირატესობას უბრალო (ანათვლებს შორის ურთიერთკავშირის გაუთვალისწინებლად) კოდირების მეთოდთან შედარებით.

მოვახდინოთ 8-ბიტის (256-დონის) ციფრული გამოსახულების სკანირება, რომლის დროსაც, მაგალითად, 10 ერთმანეთის მიმდევრო პიქსელებს გააჩნია შემდეგი დონეები:

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

თუ მოვახდენთ პიქსელების აღნიშნული დონეების უბრალოდ კოდირებას მახსოვრობის გარეშე, მაშინ გამოსახულების ყოველი პიქსელისათვის საჭირო იქნება 8 ბიტი, ანუ ათივე პიქსელისათვის – 80-ბიტის კოდური სიტყვა.

ახლა დავუშვათ, რომ კოდირების განხორციელებამდე გამოვთვალოთ სხვაობები მეზობელ პიქსელებს შორის, რის შედეგადაც მივიღებთ შემდეგი სახის მიმდევრობას:

144,	147,	150,	146,	141,	142,	138,	143,	145,	142 .
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
144,	3,	3,	- 4,	-5,	1,	-4,	5,	2,	-3 .

საწყისი მიმდევრობა სხვაობითიდან აღვიღად აღდგება უბრალო აჯამების (დისკრეტული ინტეგრირების) გზით:

144,	144+3,	147+3,	150-4,	146-5,	141+1,	142-4,	138+5,	143+2,	145-3
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
144,	147,	150,	146,	141,	142,	138,	143,	145,	142 .

ანათვლების (პიქსელების) სხვაობების მიღებული რიცხვებიდან პირველის კოდირებისათვის, ცხადია, საჭიროა 8 ბიტი (ისევე როგორც უბრალო კოდირების განხორციელებისას), ხოლო ყველა დანარჩენი რიცხვისათვის შეიძლება

გამოვიყენოთ 4-ბიტიანი სიტყვები (ერთი მათგანი ნიშნისათვის, ხოლო 3 – რიცხვის მოდულის კოდირებისათვის).

ამრიგად, კოდირების შედეგად მივიღებთ კოდურ სიტყვას, რომლის სიგრძეა $8+9+4=44$ ბიტი, ანუ თითქმის ორჯერ მოკლეს ანათვლების ინდივიდუალურ კოდირებასთან შედარებით.

დიფერენციალური კოდირების მეთოდი ფართოდ გამოიყენება იმ შემთხვევებში, როდესაც მონაცემთა ბუნება ისეთია, რომ მათი მეზობელი ანათვლები ერთმანეთისაგან უმნიშვნელოდ არიან განსხვავებული. ეს თავისებურება შეეხება ბგერით სიგნალებს, განსაკუთრებით კი სიტყვას, ასევე გამოსახულებებს, რომლებშიც მეზობელ პიქსელებს გააჩნია თითქმის ერთნაირი სიკაშკაშე. ამავდროულად უნდა ითქვას, რომ ეს მეთოდი სრულიად არ მიესადაგება ტექსტებს, ნახაზებს ან რაიმე სახის ციფრულ მონაცემებს, რომელთა მეზობელი მნიშვნელობები ერთმანეთისაგან დამოუკიდებელია.

2.8. კომპრესიის მეთოდები ინფორმაციის დანაკარგებით

როგორც უკვე იყო აღნიშნული, არსებობს მონაცემთა შეკუმშვის სისტემების ორი ტიპი:

1. სისტემები ინფორმაციის დანაკარგების გარეშე (არარღვევადი სისტემები).
2. სისტემები ინფორმაციის დანაკარგებით (რღვევადი სისტემები).

არარღვევადი კოდირების დროს საწყისი მონაცემები შეკუმშულიდან შეიძლება აღდგეს თავდაპირველი სახით, ანუ აბსოლუტურად ზუსტად. ასეთი კოდირება გამოიყენება ტექსტების, მონაცემთა ბაზების, კომპიუტერული პროგრამების და ა.შ. კომპრესიისათვის (შეკუმშვისათვის), ანუ იმ შემთხვევებში, როდესაც დაუშვებელია მათი თუნდაც უმცირესი განსხვავება. ყველა ზემოთ განხილული მეთოდი მიეკუთვნება სწორედ არარღვევად მეთოდთა ჯგუფს.

სამწუხაროდ, არარღვევადი შეკუმშვა მიუხედავად იმისა, რომ ის ძალიან მიმზიდველია საწყისი და აღდგენილი მონაცემების *აბსოლუტური თანხვედრის* გამო, გამოირჩევა დაბალი ეფექტურობით – არარღვევადი შეკუმშვის კოეფიციენტები იშვიათად აღემატება 3...5-ს (ისეთი მონაცემების კოდირების გარდა, რომლებიც გამოირჩევა ერთნაირი უბნების გამეორებადობის მაღალი ხარისხით და ა.შ.).

ამასთან ერთად უნდა აღინიშნოს, რომ ძალზე ხშირად არ არსებობს მომხმარებლისათვის საწყისი მონაცემების აბსოლუტური სიზუსტით მიწოდების აუცილებლობა. ჯერ ერთი, თვით მონაცემთა წყაროებს გააჩნია შეზღუდული დინამიური დიაპაზონი, რის გამოც ისინი საწყის მონაცემებს გამოიმუშავენ გარკვეული დონის დამახინჯებებითა და შეცდომებით. ეს დონე შეიძლება იყოს დიდი ან პატარა, მაგრამ აღწარმოების აბსოლუტური სიზუსტის მიღწევა მაინც შეუძლებელია. მეორეც, კავშირის არხებით მონაცემთა გადაცემა და მათი შენახვა ყოველთვის ხდება სხვადასხვა სახის ხელშეშლათა თანხლებით. ამიტომ მიღებული (აღწარმოებული) შეტყობინება ყოველთვის გარკვეული ხარისხით განსხვავდება გადაცემული შეტყობინებისაგან, ანუ კავშირის არხში (ან მახსოვრობის სისტემაში) ხელშეშლების არსებობისას პრაქტიკულად შეუძლებელია აბსოლუტური ზუსტი გადაცემის უზრუნველყოფა. და ბოლოს, შეტყობინებები გადაიცემა და შეინახება მიმღების მიერ მათი აღქმისა და გამოყენებისათვის. ინფორმაციის მიმღებები კი არის ადამიანის შეგრძნების ორგანოები, შემსრულებელი მექანიზმები და ა.შ. ეს უკანასკნელნიც გამოირჩევიან შეზღუდული გარჩევის უნარიანობით, ანუ ისინი ვერ ამჩნევენ უმნიშვნელო განსხვავებას ადღგენილი (აღწარმოებული) შეტყობინების აბსოლუტურად ზუსტ და მიახლოებულ მნიშვნელობებს შორის. დამახინჯებათა მიმართ მგრძობიარობის ზღვარი ასევე შეიძლება იყოს სხვადასხვა, თუმცა ის ყოველთვის არსებს.

რღვევადი კოდირება ითვალისწინებს ზემოაღნიშნულ არგუმენტებს მონაცემთა მიახლოებული ადღგენის სასარგებლოდ და შეცდომათა გარკვეული დოზით კონტროლირებადი სიდიდის ხარჯზე იძლევა შეკუმშვის ისეთი კოეფიციენტის მიღწევის საშუალებას, რომელიც ზოგჯერ ათეულობით აღემატება არარღვევადი მეთოდების გამოყენებით მიღწეულ შეკუმშვის კოეფიციენტს (შეკუმშვის ხარისხს).

რღვევადი შეკუმშვის უმრავლესობა ეფუძნება არ საკუთრივ მონაცემთა კოდირებას, არამედ მათი წრფივი გარდასახვის შედეგების (ტრანსფორმანტების), მაგალითად, ფურიეს დისკრეტული (**ფდბ**), კოსინუსური (**კბ**), ხაარის (**ხბ**), უოლშის (**უბ**) და ა.შ. გარდასახვების კოეფიციენტების (ტრანსფორმანტების) კოდირებას.

იმისათვის, რომ გავიგოთ, რას ემყარება რღვევადი შეკუმშვის სისტემის ეფექტურობა და რატომაა გარდასახვების გამოყენებით კოდირება მნიშვნელოვნად უფრო ეფექტური, ვიდრე საწყისი მონაცემების კოდირება,

სამაგალითოდ განვიხილოთ გამოსახულებათა შეკუმშვის (კომპრესიის) პოპულარული მეთოდი **JPEG** ("ჯიპეგი"), რომელიც შეიცავს რღვევადი შეკუმშვის ყველა აუცილებელ ატრიბუტს. შევნიშნოთ, რომ აღნიშნული საკითხის განხილვისას ყურადღება გამახვილებული იქნება გარდასახვებით კოდირების იდეის მიმართ და არა მათ მათემატიკურ ასპექტებზე.

აუცილებელი იქნება აგრეთვე შეკუმშვისათვის გამოყენებული წრფივი გარდასახვების არსის განხილვა, ვინაიდან მათი ფიზიკური არსის ცოდნის გარეშე ძნელია მიღებული ეფექტების მიზეზების გაგება.

2.8.1. კოდირება ბარდასახვებით. შეკუმშვის JPEG სტანდარტი

უძრავ გამოსახულებათა კოდირების პოპულარული სტანდარტი **JPEG (Joint Photographers Experts Group)** წარმოადგენს ძალზე კარგ ილუსტრაციას გარდასახვებით კოდირების საფუძველზე მიღწეული რღვევადი შეკუმშვის (კომპრესიის) პრინციპების ახსნისათვის.

გარდასახვებით კოდირების ძირითადი იდეა შეიძლება ნათელვყოთ შემდეგი მარტივი მსჯელობით. დავუშვათ, რომ საქმე გვაქვს რაიმე ციფრულ სიგნალთან (დისკრეტული ანათვლების მიმდევრობასთან). თუ თითოეული მათგანიდან გადავაგდებთ ორობითი სიმბოლოების ნახევარს (მაგალითად, რვიდან ოთხს), მაშინ *ორჯერ შემცირდება კოდის სიჩქარე R და დაიკარგება სიგნალში არსებული ინფორმაციის ნახევარი*.

თუ სიგნალს დავამუშავებთ ფურიეს (ან რომელიმე სხვა) გარდასახვით, დავეყოფთ მას ორ დაბალ და მაღალ სიხშირულ შემდგენებად, მოვახდენთ დისკრეტიზაციას, თითოეულ მათგანს დავეკვანტავთ და მხოლოდ მაღალსიხშირული შემდგენების ორობითი თანრიგების ნახევარს გადავაგდებთ, მაშინ *კოდის საბოლოო სიჩქარე ერთი მესამედით შემცირდება, ხოლო ინფორმაციის დანაკარგი იქნება სულ რაღაც 5%*.

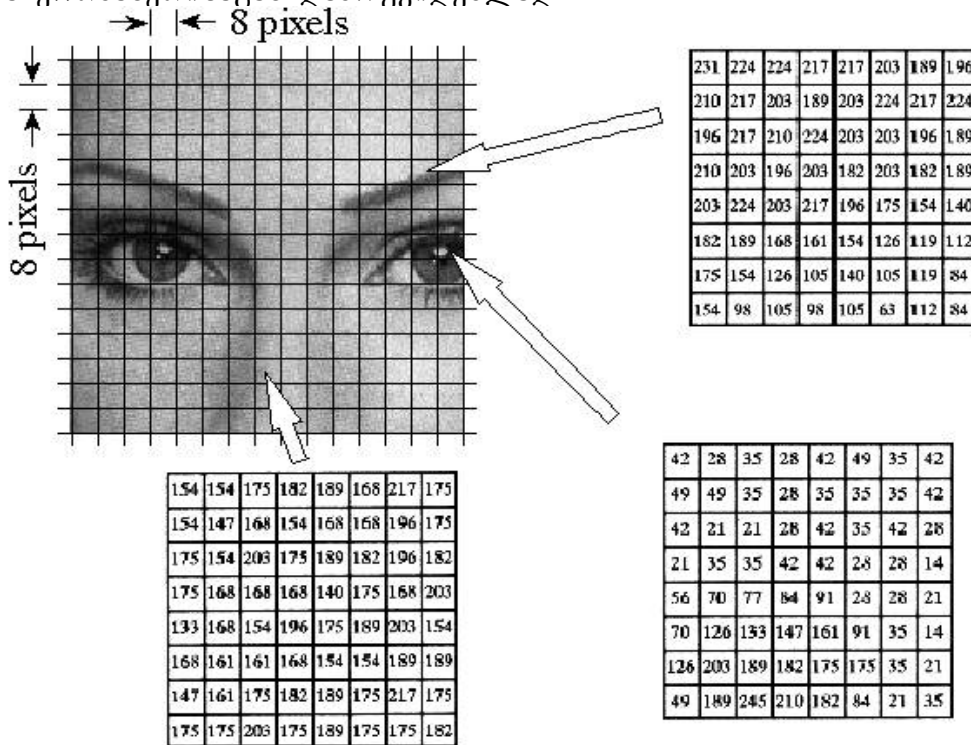
ეს ეფექტი იმითაა განპირობებული, რომ უმრავლესი სიგნალების დაბალსიხშირული შემდგენები (მსხვილი დეტალები) ჩვეულებრივ გაცილებით ინტენსიურად გვხვდება და ისინი შეიცავენ გაცილებით მეტ ინფორმაციას, ვიდრე მაღალსიხშირული შემდგენები (წვრილი დეტალები). ყოველივე აღნიშნული თანაბარი დოზით შეეხება როგორც ბგერით სიგნალებს, ასევე გამოსახულებებს.

განვიხილოთ შეკუმშვის **JPEG** ალგორითმი იმ შავ-თეთრი გამოსახულების კოდირების შემთხვევაში, რომელიც წარმოადგენს 256-დონიანი (8 ორობითი

სიმბოლო) ანათვლების (პიქსელების) ნაკრებს. ეს ყველაზე გავრცელებული მეთოდია გამოსახულებათა შენახვისათვის (არქივირებისათვის) – ეკრანზე ყოველ წერტილს შეესაბამება 1 ბაიტი (8 ბიტი – 256 შესაძლო მნიშვნელობები), რომელიც განსაზღვრავს მის სიკაშკაშეს. 255 – მაქსიმალური სიკაშკაშე (თეთრი ფერი), 0 – მინიმალური სიკაშკაშე (შავი ფერი). შუალედური მნიშვნელობები კი ადგენენ რუხი ფერების მთელ დანარჩენ გამას (ნახ. 2.8).

შეკუმშვის **JPEG** ალგორითმი იწყება გამოსახულების დაყოფით **8x8** ზომის კვადრატულ ბლოკებად (8x8=64 პიქსელი). რატომ მაინც და მაინც 8x8 და არა 2x2 ან 32x32? ბლოკის ასეთი ზომის შერჩევა იმითაა განპირობებული, რომ მისი უფრო მცირე ზომისას კოდირების ეფექტი იქნება მცირე (1x1 ზომის შემთხვევაში ეფექტი საერთოდ არ არსებობს), ხოლო დიდი ზომის ბლოკებისას (ფრაგმენტებისას) ბლოკის ფარგლებში გამოსახულების თვისებები ხლიერ ცვალებადი აღმოჩნდება, რის გამოც კვლავ შემცირდება კოდირების ეფექტურობა.

2.8 ნახაზზე ნაჩვენებია რამდენიმე ასეთი ბლოკი (ციფრული ანათვლების მატრიცების სახით), რომლებიც აღებულია გამოსახულების სხვადასხვა უბნებიდან. შემდგომში ასეთნაირად ფორმირებული ბლოკები დამუშავდება და კოდირდება ერთმანეთისაგან დამოუკიდებლად.



ნახ. 2.8. გამოსახულების დაყოფა (დანაწევრება) 8x8 ზომის ბლოკებად

შეკუმშვის მეორე ეტაპია ყოველი ბლოკის მიმართ დისკრეტული კოსინუსური გარდასახვის (ԶԾ) გამოყენება. მონაცემთა შეკუმშვისათვის ცდილობდნენ სხვადასხვა გარდასახვების, მათ შორის ამისათვის სპეციალურად შექმნილი კარუნენ-ლოევის გარდასახვის, გამოყენებას, რომელიც უზრუნველყოფს შეკუმშვის მაქსიმალურად შესაძლო კოეფიციენტის მიღებას. მაგრამ ეს უკანასკნელი პრაქტიკაში ძნელად რეალიზებადი აღმოჩნდა. ფურიეს გარდასახვა (ՅԾ) ხორციელდება ძალზე მარტივად, მაგრამ იგი ვერ უზრუნველყოფს კარგ შეკუმშვას. ამის გამო ყურადღება შეჩერდა დისკრეტულ კოსინუსურ გარდასახვაზე, რომელიც წარმოადგენს ՅԾ-ის ნაირსახეობას. ՅԾ-გან განსხვავებით, რომელიც სიგნალის განშლისათვის იყენებს სინუსურ და კოსინუსურ სიხშირულ შემდგენებს, ԶԾ-ში გამოიყენება მხოლოდ კოსინუსური შემდგენები. დისკრეტული კოსინუსური გარდასახვა, ისევე როგორც ნებისმიერი სხვა წრფივი გარდასახვა, იძლევა გამოსახულებათა სივრცული წარმოდგენიდან (ანათვლების ანუ პიქსელების ნაკრები) სპექტრალურ წარმოდგენაზე (სიხშირული შემდგენების ნაკრები) გადასვლის საშუალებას და პირიქით.

$IMG(x,y)$ გამოსახულების დისკრეტული კოსინუსური გარდასახვა შეიძლება წარმოვადგინოთ შემდეგი სახით:

$$DCT(u,v) = \sqrt{2/N} \sum_{ij} IMG(x_i, y_j) \cos((2i+1)\pi u/2N) \cos((2j+1)\pi v/2N), \quad (2.17)$$

სადაც $N=8$, $0 < i < 7$, $0 < j < 7$, ანუ მატრიცულ ფორმაში

$$RES = DCT^T \cdot IMG \cdot DCT, \quad (2.18)$$

სადაც DCT – 8×8 ზომის გარდასახვისათვის ბაზისური (კოსინუსური) კოეფიციენტების მატრიცაა, რომელსაც აქვს შემდეგი სახე:

$$DCT = \begin{pmatrix} .353553 & .353553 & .353553 & .353553 & .353553 & .353553 & .353553 & .353553 \\ .490393 & .415818 & .277992 & .097887 & -.097106 & -.277329 & -.415375 & -.490246 \\ .461978 & .191618 & -.190882 & -.461673 & -.462282 & -.192353 & .190145 & .461366 \\ .414818 & -.097106 & -.490246 & -.278653 & .276667 & .490710 & .099448 & -.414486 \\ .353694 & -.353131 & -.354256 & .352567 & .354819 & -.352001 & -.355378 & .351435 \\ .277992 & -.490246 & .096324 & .416700 & -.414486 & -.100228 & .491013 & -.274673 \\ .191618 & -.462282 & .461366 & -.189409 & -.193822 & .463187 & -.460440 & .187195 \\ .097887 & -.278653 & .416700 & -.490862 & .489771 & -.413593 & .274008 & -.092414 \end{pmatrix} \quad (2.19)$$

ამრიგად, გამოსახულების 8×8 ზომის (პიქსელების რაოდენობაა 64) ბლოკისადმი დისკრეტული კოსინუსური გარდასახვის გამოყენების შედეგად

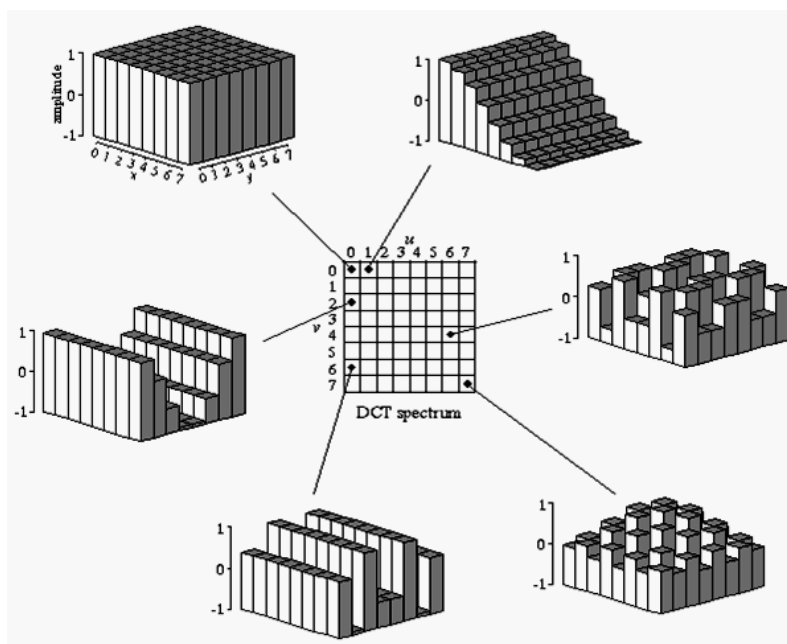
მივიღებთ ორგანზომნილებიან სპექტრს, რომლის ზომაცაა ასევე **8x8**. სხვა სიტყვებით რომ ვთქვათ, გამოსახულების ანათვლების წარმოდგენი **64** რიცხვი გადაიქცევა გამოსახულების ამ ბლოკის დისკრეტული კოსინუსური გარდასახვის სპექტრის **64** ანათვლად.

გავიხსენოთ, თუ რას წარმოადგენს სიგნალის სპექტრი. ესაა კოეფიციენტების სიდიდეები, რომელთანაც ერთად შესაბამისი სპექტრული შემდგენები შედიან იმ ჯამის შემადგენლობაში, რომელიც საბოლოოდ იძლევა ამ სიგნალს. ძირითად სპექტრულ შემდგენებს, რომლებადაც იშლება სიგნალი, ხშირად ბაზისურ ფუნქციებს უწოდებენ. ფურიეს გარდასახვისათვის ბაზისური ფუნქციებია სხვადასხვა სიხშირის სინუსები და კოსინუსები.

ბლოკების **8x8** ზომის შემთხვევაში დისკრეტული კოსინუსური გარდასახვის ბაზისური ფუნქციები მოცემულია შემდეგი ფორმულით:

$$b[x, y] = \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right], \quad (2.20)$$

ხოლო თავად ბაზისური ფუნქციები წარმოდგენილია 2.9 ნახაზზე.



ნახ. 2.9. დისკრეტული კოსინუსური გარდასახვის (**DCB—DCT**) ბაზისური ფუნქციები

(0,0) ინდექსების შესაბამისი ყველაზე დაბალსიხშირული ბაზისური ფუნქცია გამოსახულია ნახაზის ზედა მარცხენა კუთხეში, ხოლო ყველაზე

მაღალსიხშირული – ქვედა მარჯვენა კუთხეში. ბაზისური ფუნქცია $(0,1)$ წარმოადგენს ერთი კოორდინატის მიხედვით კოსინუსოიდის ნახევარს, ხოლო მეორე კოორდინატის მიხედვით – მუდმივს (კონსტანტას), ხოლო ბაზისური ფუნქცია $(1,0)$ კი წარმოადგენს იმავეს, ოღონდ მობრუნებულს 90° -ით.

დისკრეტული კოსინუსური გარდასახვა გამოითვლება გამოსახულების $8x8$ ზომის ბლოკების პიქსელების საბაზო ფუნქციებიდან თითოეულზე ელემენტური გადამრავლებითა და აჯამებით. ამის შედეგად, მაგალითად, დისკრეტული კოსინუსური გარდასახვის $(0,0)$ ინდექსებიანი კომპონენტი იქნება გამოსახულების ბლოკის ყველა ელემენტის ჯამი, ანუ ბლოკის საშუალო სიკაშკაშე. $(0,1)$ ინდექსებიანი კომპონენტში ერთნაირი წონით საშუალოვდება გამოსახულების ბლოკის ყველა ჰორიზონტალური ელემენტი (დეტალი), ხოლო ვერტიკალის მიხედვით უდიდესი წონა ენიჭება გამოსახულების ბლოკის ზედა ნაწილის ელემენტებს და ა.შ. შეიძლება შევნიშნოთ, რომ დისკრეტული კოსინუსური გარდასახვის მატრიცაში რაც უფრო ქვევით და მარჯვნივაა მოთავსებული მისი კომპონენტი, გამოსახულების მით უფრო მაღალსიხშირულ დეტალებს შეესაბამება ის.

დისკრეტული კოსინუსური გარდასახვის სპექტრიდან საწყისი გამოსახულების მისაღებად (უკუგარდაქმნის შესრულება), საჭიროა $(0,0)$ ინდექსებიანი საბაზო ფუნქცია გამრავლდეს $(0,0)$ კოორდინატების მქონე სპექტრულ კომპონენტზე, შემდეგ მას დაემატოს $(1,0)$ საბაზო ფუნქციის ნამრავლი სპექტრის $(1,0)$ კოორდინატის მქონე კომპონენტზე და ა.შ.

2.9 ცხრილში ნაჩვენებია გამოსახულების ერთერთი ბლოკის ელემენტებისა და მისი $8x8$ -ის სპექტრის კომპონენტების (ტრანსფორმანტების, კოეფიციენტების) ციფრული მნიშვნელობები.

თუ ყურადღებით შევხედავთ $8x8$ -ის შედეგად მიღებულ კოეფიციენტებს (კომპონენტებს, ტრანსფორმანტებს), მაშინ შევნიშნავთ, რომ მათი საკმაოდ დიდი რაოდენობა ნულის ტოლი ან ძალიან მცირე სიდიდისაა (1 – 2). ესენი მაღალსიხშირული შემდგენებია, რომლებიც (ჩვეულებრივ) შეიძლება გადავავლოთ (მოვიცილოთ) ან დავამრგვალოთ უახლოეს მთელ მნიშვნელობამდე.

აღვნიშნოთ $8x8$ -ის სპექტრის ძალზე საინტერესო თავისებურება: მისი უდიდესი მნიშვნელობები (კომპონენტები) თავმოყრილია 2.9 ცხრილის მარცხენა ზედა კუთხეში (დაბალსიხშირული შემდგენები), მარჯვენა ქვედა ნაწილი (მაღალსიხშირული შემდგენები) კი შევსებულია შედარებით პატარა ციფრებით.

თუმცა სპექტრში ციფრების საერთო რაოდენობა იგივეა, რაც გამოსახულების ბლოკში ($8 \times 8 = 64$), ანუ ჯერჯერობით არავითარი შეკუმშვა (კომპრესია) არ მომხდარა და თუ შევასრულებთ უკუგარდაქმნას, მივიღებთ გამოსახულების იმავე ბლოკს.

ცხრილი 2.9

საწყისი მონაცემები							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15
DCT-ის შედეგები (კომპონენტები)							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

JPEG ალგორითმის მუშაობის შემდეგი ეტაპია დაკვანტვა (ცხრილი 2.10).

დაკვანტვა გულისხმობს **DCT**-ის თითოეული კოეფიციენტის გაყოფას რაიმე ციფრზე დაკვანტვის მატრიცის მიხედვით. ეს მატრიცა შეიძლება იყოს ფიქსირებული ან უფრო ხარისხიანი და ევექტური შეკუმშვის (კომპრესიის) მიღების მიზნით აიგოს (ფორმირდეს) საწყისი სურათის ანალიზის საფუძველზე. რაც უფრო მეტია რიცხვები, რომლებზეც იყოფა კომპონენტები, მით მეტი იქნება გაყოფის შედეგად მიღებული ნულოვანი მნიშვნელობები და, შედეგად, მით მეტი იქნება კომპრესია და შესამჩნევი დანაკარგები.

<i>დკბ-ის ადრე მიღებული შედეგები (კომპონენტები)</i>							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4
<i>დაკვანტვის ცხრილი</i>							
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99
<i>დაკვანტვის შედეგი</i>							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

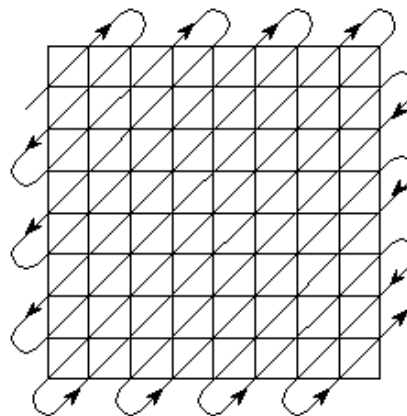
სრულიად ნათელია, რომ დაკვანტვის ცხრილის შერჩევაზე მნიშვნელოვნადაა დამოკიდებული როგორც შეკუმშვის (კომპრესიის) ეფექტურობა (ნულების რაოდენობა დაკვანტულ სპექტრში), ასევე აღდგენილი სურათის ხარისხი.

ამრიგად, ჩვენ დავამრგვალებთ *დკბ*-ის შედეგები და მივიღეთ გამოსახულების ამა თუ იმ ხარისხით დამახინჯებული ბლოკური სპექტრი.

კოდირების *JPEG* ალგორითმის მუშაობის შემდეგი ეტაპია *დკბ*-ის სპექტრის **8x8** ზომის მატრიცის გადასახვა წრფივ მიმდევრობად ისე, რომ შექმნილი დაგვარად მოვახდინოთ სპექტრის შემდგენების ერთად დაჯგუფება

ყველა დიდი და ყველა ნულოვანი მნიშვნელობების მიხედვით. ნათელია, რომ ამისათვის საჭიროა *დკბ*-ის მატრიცის კომპონენტების (კოეფიციენტების) წაკითხვა 2.10 ნახაზზე ნაჩვენები მიმდევრობით, ანუ ზიგზაგისებურად ზედა მარცხენა კუთხიდან ქვედა მარჯვენა კუთხისაკენ. ამ პროცედურას *ზიგზაგ-სკანირება*ს უწოდებენ.

ასეთი გარდაქმნის შედეგად *დკბ*-ის დაკვანტული კოეფიციენტების 8×8 ზომის მატრიცა გარდაიქმნება 64 რიცხვისაგან შედგენილ წრფივ მიმდევრობად, რომელთაგან უდიდესი ნაწილია ერთმანეთს მიყოლებული ნულები. ცნობილია, რომ ასეთი ნაკადები შეიძლება შევკუმშოთ ძალზე ეფექტურად *გამეორებათა სიგრძეების კოდირების* მეთოდის საშუალებით, რაც გათვალისწინებულია *JPEG* ალგორითმში.



ნახ. 2.10. ზიგზაგ-სკანირება

JPEG-კოდირების მომდევნო მეხუთე ეტაპზე ხდება ნულების მიღებული მიმდევრობის კოდირება ხდება *გამეორებათა სიგრძეების* მეთოდის საფუძველზე.

და ბოლოს, *JPEG* ალგორითმის მუშაობის ბოლო ეტაპია მიღებული მიმდევრობების კოდირება რომელიმე სტატისტიკური ალგორითმის გამოყენებით. ჩვეულებრივ გამოიყენებენ არითმეტიკულ კოდირებას ან ხაფმენის ალგორითმს, რის შედეგადაც დებულობენ ახალ მიმდევრობას, რომლის ზომაც გაცილებით ნაკლებია საწყისი მონაცემების ზომასთან შედარებით.

კოდირების ბოლო ორ ეტაპს ჩვეულებრივ უწოდებენ მეორად შეკუმშვას. სწორედ ამ ეტაპზე ხდება არარღვევადი სტატისტიკური კოდირება და მონაცემების მოცულობის მნიშვნელოვანი შემცირება მათთვის დამახასიათებელი სტრუქტურის გათვალისწინებით.

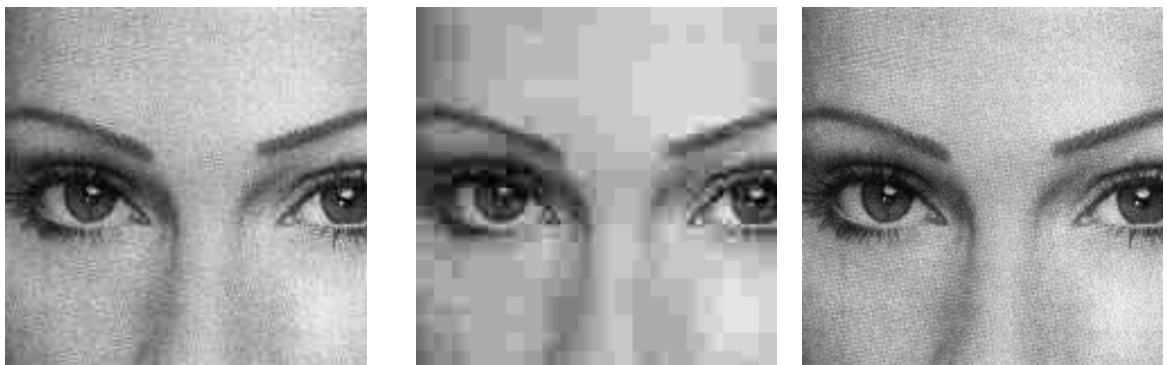
JPEG ალგორითმის საფუძველზე შეკუმშული მონაცემების დეკოდირება ხდება ზუსტად ისევე, როგორც კოდირება, მაგრამ ამ შემთხვევაში ყველა ოპერაცია სრულდება უკუმიმდევრობით.

ხაფმენის (ან *LZW* ან არითმეტიკული კოდირების) არარღვევადი უკუკოდირებისა (ეგრეთ წოდებული განფუთვის) და წრფივი მიმდევრობების 8×8 ზომის ბლოკებად დალაგების შემდეგ ხდება *სპექტრული კომპონენტების დეკვანტირება*

დაკვანტვის ცხრილის კოდირების დროს შენარჩუნებული კომპონენტების საშუალებით. ამისათვის *DCB*-ის დალაგებული 64 მნიშვნელობა მრავლდება ცხრილის შესაბამის რიცხვებზე. ამის შემდეგ ყოველ ბლოკს გარდაქმნიან უკუკოსინუსური გარდასახვით, რომლის შესაბამისი პროცედურა იგივეა, რაც პირდაპირი გარდასახვის პროცედურა, ოღონდ გასწვავება მხოლოდ გარდასახვის მატრიცის ნიშნებშია. დეკოდირებისას მოქმედებათა თანმიმდევრობა და მიღებული შედეგები წარმოდგენილია 2.11 ცხრილში.

ცხადია, რომ აღდგენილი მონაცემები გარკვეულწილად განსხვავდება საწყისი მონაცემებისაგან. ეს ბუნებრივია იმიტომ, რომ *JPEG* დამუშავდა სწორედ დანაკარგებით შეკუმშვის მიზნით.

2.11 ნახაზზე წარმოდგენილია საწყისი გამოსახულება (მარჯვნივ) და ასევე *JPEG* ალგორითმის მიხედვით 10-ჯერ (მარცხნივ) და 45-ჯერ (ცენტრში) შეკუმშული გამოსახულებები. უკანასკნელ შემთხვევაში ხარისხის გაუარესება სრულიად შესაძლებელია, თუმცა ასევე ცხადია მოცულობაში მოგებაც.



ნახ. 2.11. საწყისი და *JPEG* ალგორითმით კოდირებული გამოსახულებები

ცხრილი 2.11

დაკვანტული მონაცემები							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

<i>დეკანტირებული მონაცემები</i>							
240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
<i>უკუ ღებ-ის შედეგები</i>							
144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158
<i>საწყისი მონაცემები (შედარებისათვის)</i>							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15

ამრიგად, **JPEG**–შეკუმშვა შედეგება შემდეგი ეტაპებისგან:

1. გამოსახულების დაყოფა (დანაწევრება) **8x8** ზომის ბლოკებად.
2. თითოეული ბლოკისადმი დისკრეტული კოსინუსური გარდასახვის გამოყენება.
3. **ღებ**-ის კოეფიციენტების დამრგვალება წონითი კოეფიციენტების მოცემული მატრიცის შესაბამისად.
4. დამრგვალებული **ღებ**-ის მატრიცის გარდაქმნა წრფივ მიმდევრობად ზიგზაგისებური სკანირებით (წაკითხვით).
5. გამეორებათა კოდირება ნულოვანი კომპონენტების რიცხვის შემცირების მიზნით.

6. მიღებული შედეგის სტატისტიკური კოდირება ხაფმენის ან არითმეტიკული კოდის გამოყენებით.

დეკოდირება ხდება ზუსტად ისევე, როგორც კოდირება, ოღონდ უკუმიმართულებით.

შეკუმშვის **JPEG** ალგორითმის მნიშვნელოვანი ღირსებებია:

– შეკუმშვის ხარისხის ფართო საზღვრებში (2–დან 200–დე) მიღწევის შესაძლებლობა;

– ნებისმიერი თანრიგიანობის მქონე გამოსახულებების დამუშავების შესაძლებლობა;

– პირდაპირი და უკუგარდაქმნების პროცედურების სიმეტრიულობა.

ნაკლოვანებებს შეიძლება მივაკუთვნოთ ფერების მკვეთრი გადასვლის უბნებზე შარავანდის (გიბსის ეფექტის) არსებობა და შეკუმშვის მაღალი ხარისხის შემთხვევაში გამოსახულების **8x8** ზომის ფრაგმენტებად დანაწევრების შესაძინეადობა.

2.8.2. ფრაქტალური მეთოდი

ფრაქტალური შეკუმშვა ემყარება იმას, რომ გამოსახულების უფრო კომპაქტური წარმოდგენა ხდება იტერირებადი ფუნქციების სისტემის (**Iterated Function System — IFS**) კოეფიციენტების საშუალებით.

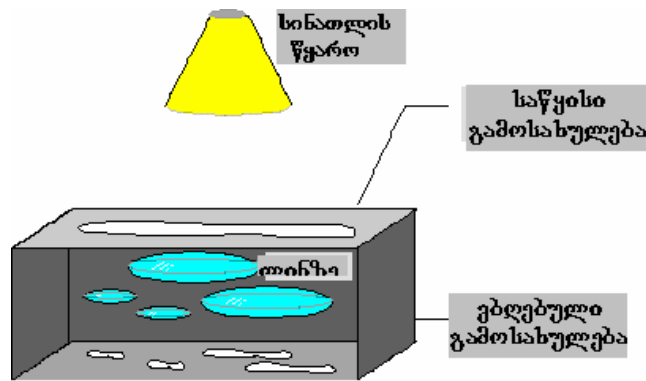
მანამ, სანამ განვიხილავთ თვით ფრაქტალური შეკუმშვის პროცესს, გავარკვიოთ, თუ როგორ აგებს გამოსახულებას **IFS**, ანუ განვიხილოთ დეკომპრესიის პროცესი.

ზუსტად რომ ვთქვათ, **IFS** წარმოადგენს სამგანზომილებიანი აფინური გარდასახვების ნაკრებს, რომელიც ჩვენს შემთხვევაში ერთ გამოსახულებას გადაიყვანს მეორეში. გარდაქმნას ექვემდებარებიან სამგანზომილებიანი სივრცის წერტილები (გამოსახულების წერტილის **X** და **Y** კოორდინატები და წერტილის **I** სიკაშკაშე).

ეს პროცესი გამარტივებული სახით შეიძლება წარმოვადგინოთ შემდეგნაირად: განვიხილოთ ფოტომაკოპირებელი მანქანა (ნახ. 2.12), რომელიც შედგება ეკრანისაგან, რომელზეც საწყისი გამოსახულებაა წარმოდგენილი, და

ლინზების სისტემისაგან, რომელიც გამოსახულებას ასახავს (აპროექციებს) მეორე ეკრანზე. იგი შეიძლება ასრულებდეს შემდეგ მოქმედებებს:

- ლინზებს შეუძლიათ დააპროექციოს გამოსახულების ნებისმიერი ფორმის ნაწილი ახალი გამოსახულების სხვა ნებისმიერ ადგილას;
- არ იკვეთებიან არეები, რომლებშიც პროექცირდება გამოსახულებები;
- ლინზას შეუძლია სიკაშკაშის შეცვლა და კონტრასტულობის შემცირება;
- ლინზას შეუძლია გამოსახულების თავისი ფრაგმენტის სარკისებური არეკვლა და მისი მობრუნება;
- ლინზამ უნდა დაამასშტაბიროს (შეამციროს) გამოსახულების თავისი ფრაგმენტი.



ნახ. 2.12. ფოტომაკოპირებელი მანქანა

ლინზების განლაგებისა და მათი მახასიათებლების რეგულირების საშუალებით შეიძლება მიღებული გამოსახულების მართვა. მანქანის მუშაობის ერთი იტერაცია იმაში მდგომარეობს, რომ საწყისი გამოსახულების საშუალებით პროექცირების გზით აიგება ახალი გამოსახულება, რის შემდეგაც ეს უკანასკნელი მიიჩნევა როგორც საწყისი გამოსახულება. მტკიცდება, რომ იტერაციების სედეგად ჩვენ მივიღებთ გამოსახულებას, რომელიც შეწყვეტს ცვლილებას. იგი დამოკიდებული იქნება მხოლოდ ლინზების განლაგებასა და მახასიათებლებზე და არ იქნება დამოკიდებული საწყის სურათზე. ამ გამოსახულებას "უძრავ წერტილს" ან მოცემული *IFS*-ის ატრაქტორს უწოდებენ. შესაბამისი თეორია იძლევა ყოველი *IFS*-ისათვის ზუსტად ერთი უძრავი წერტილის არსებობის გარანტიას.

იმის გამო, რომ ლინზების ასახვა შემკუმშველია, ყოველი ლინზა ჩვენს გამოსახულებაში ნათლად იძლევა თვითმსგავს (თავის თავის მსგავს) არეებს.

თვითმსგავსობის გამო ვლელულობთ გამოსახულების რთულ სტრუქტურას მისი ნებისმიერი გადიდებისას. ამრიგად, ინტუიციურად გასაგებია, რომ მაიტერირებელი ფუნქციების სისტემა იძლევა *ფრაქტალს* – თვითმსგავს მათემატიკურ ობიექტს.

IFS–ის დახმარებით მიღებული გამოსახულებებიდან ყველაზე ცნობილია ეგრეთ წოდებული "ბარნსლის გვიმრა" (ნახ. 2.13), რომელიც მოცემულია ოთხი აფინური გარდასახვით (ანუ, ჩვენი ტერმინოლოგიით, "ლინზებით"). ყოველი გარდასახვა კოდირდება ზუსტად თვლადი ბაიტებით მაშინ, როდესაც მათი დახმარებით აგებულ გამოსახულებას შეიძლება დაჭირდეს რამდენიმე მეგაბაიტი.



ნახ. 2.13. გამოსახულება "ბარნსლის გვიმრა".

ამ გამოსახულებაზე შეიძლება ოთხი არის გამოყოფა, რომელთა გაერთიანება მოიცავს მთელ გამოსახულებას და რომელთაგან თითოეული მათგანი მსგავსია მთლიანი გამოსახულებისა (არ დაგავიწყდეთ გვიმრის ღეროს შესახებ).

ზემოაღნიშნულიდან გამომდინარე ნათელი ხდება ფრაქტალური კოდერის მუშაობის პრინციპი. ასევე ცხადია, რომ შეკუმშვისათვის კოდერს დაჭირდება საკმაოდ დიდი დრო.

ფაქტიურად ფრაქტალური კომპრესია (შეკუმშვა) – ესაა გამოსახულებაში თვითმსგავსი არეების ძებნა და მათთვის აფინური გარდასახვების პარამეტრების განსაზღვრა. ამისათვის საჭიროა გამოსახულების სხვადასხვა ზომის ფრაგმენტების გადარჩევისა და მათი ყველა შესაძლო ვარიანტის შედარების განხორციელება. მცირე ზომის გამოსახულებებისთვისაც კი დისკრეტულობის გათვალისწინების შემთხვევაში ჩვენ მივიღებთ გადასრჩევი ვარიანტების ასტრონომიულად დიდ რაოდენობას, ამასთან გარდასახვების კლასების

მკვეთრად შემცირებაც (მაგალითად, მასშტაბირების ხარჯზე) კი არ იძლევა დროის მნიშვნელოვან მოგებას. გარდა ამისა, იკარგება გამოსახულების ხარისხი. ფრაქტალური კომპრესიის მიმართულებით ჩატარებულ კვლევათა უდიდესი ნაწილი მიმართულია არქივაციის დროის შემცირებისაკენ, რომელიც საჭიროა ხარისხიანი გამოსახულების მისაღებად.

2.8.3. რეკურსიული (ტალღური) ალგორითმი

რეკურსიული შეკუმშვის ინგლისურენოვანი დასახელებაა *wavelet* ("ვეივლეტი"). იგი ითარგმნება როგორც ტალღური შეკუმშვა და შეკუმშვა ნახტომების გამოყენებით. დიდი ხანია ცნობილია არქივაციის ეს სახეობა და იგი პირდაპირ გამომდინარეობს არეების კოპრენტულობის გამოყენების იდეიდან. ალგორითმი ორიენტირებულია მდორე გადასვლების მქონე შავ-თეთრ და ფერად გამოსახულებებზე და იგი იდეალურია რენტგენის სურათებისათვის. შეკუმშვის კოეფიციენტი მოცემულია 5–100-ის ფარგლებში. შეკუმშვის უფრო დიდი კოეფიციენტის აღებისას მკვეთრი გადასვლების საზღვრებზე, განსაკუთრებით თუ ისინი დიაგნოსტიკური მიმართულებისაა, გამოსახულების რამდენიმე პიქსელის (ელემენტის) ფარგლებში წარმოიქმნება "კიბისებური ეფექტი" – სხვადასხვა სიკაშკაშის საფეხურები.

ალგორითმის იდეა მდგომარეობს იმაში, რომ *საკუთრივ გამოსახულებათა კოდირების ნაცვლად ინახება გამოსახულების მეზობელი ბლოკების საშუალო მნიშვნელობათა შორის სხვაობა*, რომელიც ჩვეულებრივ ნულთან ახლოსაა.

ორი a_{2i} და a_{2i+1} რიცხვი ყოველთვის შეიძლება წარმოვადგინოთ $b^1_i = (a_{2i} + a_{2i+1})/2$ და $b^2_i = (a_{2i} - a_{2i+1})/2$ სახით. ამის მსგავსად a_i მიმდევრობა შეიძლება წყვილ-წყვილად გარდაიქმნას $b^{1,2}_i$ მიმდევრობად.

განვიხილოთ მაგალითი. ვთქვათ ვკუმშავთ a_i პიქსელების შემდეგი რვა მნიშვნელობისაგან შედგენილ სტრიქონს: (220, 211, 212, 218, 217, 214, 210, 202). მაშინ მივიღებთ b_{1i} და b_{2i} მიმდევრობებს: (215.5, 215, 215.5, 206) და (4.5, -3, 1.5, 4). შევნიშნოთ, რომ b_{2i} -ის მნიშვნელობები საკმაოდ ახლოსაა ნულთან. ახლა b_{1i} წარმოვიდგინოთ როგორც a_i და გავიმეოროთ ოპერაცია. ეს მოქმედება ხორციელდება რეკურსიულად, საიდანაც გამომდინარეობს ალგორითმის დასახელება. (215.5, 215, 215.5, 206)–დან მივიღებთ (215.25, 210.75) და (0.25, 4.75) მიმდევრობებს. თუ მიღებულ კოეფიციენტებს დავამრგვალებთ და შევკუმშავთ, მაგალითად,

ხაფმენის კოდით, მაშინ მივიღებთ კოდირების შედეგს. შევნიშნოთ, რომ ჩვენ გარდასახვა გამოვიყენეთ მხოლოდ ორჯერ. რეალურად კი შეიძლება *wavelet*-გარდასახვის 4-6-ჯერ გამოყენება, რითაც მიიღწევა მნიშვნელოვანი შეკუმშვა (კომპრესია).

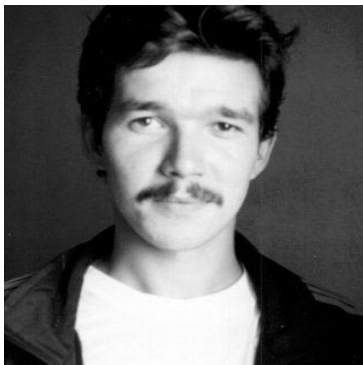
ანალოგიურად ხდება ალგორითმის რეალიზაცია ორგანზომილებიანი მონაცემებისათვის.

თუ ჩვენ გვაქვს ოთხი წერტილისაგან შედგენილი კვადრანტი, რომელთა სიკაშკაშეებია $a_{2i,2j}$, $a_{2i+1,2j}$, $a_{2i,2j+1}$ და $a_{2i+1,2j+1}$, მაშინ

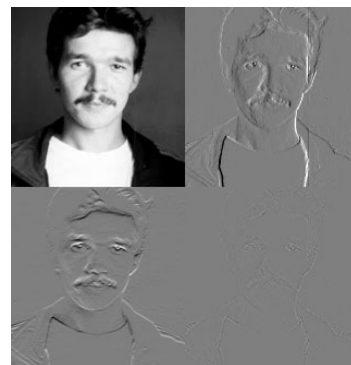
$$\begin{aligned} b_{i,j}^1 &= (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1}) / 4, \\ b_{i,j}^2 &= (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1}) / 4, \\ b_{i,j}^3 &= (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1}) / 4, \\ b_{i,j}^4 &= (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1}) / 4. \end{aligned} \quad (2.21)$$

512x512 ზომის გამოსახულებისათვის ამ ფორმულების გამოყენებით პირველი გარდასახვის შემდეგ მივიღებთ **256x256** ზომის ოთხ მატრიცას (ნა. 2.14 და ნახ. 2.15).

საწყისი გამოსახულება	B^1	B^2
	B^3	B^4



ნახ. 2.14. **512x512** ზომის გამოსახულება



ნახ. 2.15. **256x256** ზომის ოთხი გამოსახულება

ადვილი მისახვედრია, რომ პირველად შეინახება გამოსახულების შემცირებული ასლი, მეორედ – პიქსელების მნიშვნელობათა წყვილების გასაშუალებული სხვაობები პორიზონტალური მიმართულებით, მესამედ – პიქსელების მნიშვნელობათა წყვილების გასაშუალებული სხვაობები ვერტიკალური მიმართულებით, ხოლო მეოთხედ – პიქსელების გასაშუალებული სხვაობები დიაგონალური მიმართულებით.

ამის მსგავსად გარდასახვა შეიძლება გავიმეოროთ და პირველი მატრიცის მაგივრად მივიღოთ **128x128** ზომის ოთხი მატრიცა.

გარდასახვის მესამედ გამეორების შემთხვევაში საბოლოოდ მივიღებთ **64x64** ზომის ოთხ მატრიცას, **128x128** ზომის სამ მატრიცას და **256x256** ზომის სამ მატრიცას. შემდგომი შეკუმშვა ხდება იმის გათვალისწინებით, რომ სხვაობით მატრიცებში ბევრია ნულის ტოლი ან ნულთან ახლოს მყოფი მნიშვნელობები, რომლებიც დაკავანტვის შემდეგ შეიძლება ეფექტურად შევკუმშოთ.

ამ ალგორითმის ღირსებებს შეიძლება მივაკუთვნოთ გამოსახულების ქსელში გადაცემისას მისი თანდათანობითი "გამჟღავნების" შესაძლებლობის მარტივი რეალიზაცია. გარდა ამისა, ვინაიდან გამოსახულების დასაწყისში ჩვენ ფაქტიურად ვინახავთ მის შემცირებულ ასლს, ამიტომ მარტივდება "გაუხეშებელი" გამოსახულების ჩვენება.

მოცემული მეთოდი, **JPEG**–სა და ფრაქტალური ალგორითმისაგან განსხვავებით, არაა ამუშავებს **8x8** ზომის ბლოკებს. უფრო ზუსტად კი, ამუშავებს **2x2**, **4x2**, **8x8** და ა.შ. ზომის ბლოკებს. თუმცა იმის გამო, რომ ამ ბლოკების კოეფიციენტები დამოუკიდებლად შეინახება, ადვილად შეიძლება ავიცილოთ თავიდან გამოსახულების "მოზაიკურ" კვადრატებად დანაწევრების ეფექტი.

2.9. მოძრავ გამოსახულებათა (ვიდეო) კომპრესიის მეთოდები

მოძრავ გამოსახულებებთან მუშაობისას ძირითადი პრობლემაა მონაცემთა დიდი მოცულობის არსებობა. მაგალითად, კომპაქტ-დისკზე საშუალო ხარისხით ჩაწერისას მასში შეიძლება ჩაეტიოს რამდენიმე ათასი ფოტო, 10 სთ-ზე მეტი ხანგრძლივობის მუსიკა და მხოლოდ ნახევარი სთ-ის ხანგრძლივობის ვიდეო. სატელევიზიო ფორმატის ვიდეო **720x720** წერტილი კადრში და **25** კადრი წუთში **RGB** სისტემაში მოითხოვს დაახლოებით 240 მბიტი/წმ ნაკადს (1,8 გბიტი/წთ-ში). ამასთან შეკუმშვის ჩვეულებრივი მეთოდები, რომლებიც ორიენტირებულია ცალკეული კადრების (უძრავი გამოსახულებების) კოდირებაზე, ვერ ხსნიან პრობლემას, რადგან ბიტების ნაკადის **10–20**-ჯერ შემცირების შემთხვევაშიც კი იგი მაინც პრაქტიკული გამოყენებისათვის რჩება ძალიან დიდი.

მოძრავი გამოსახულებების შეკუმშვისას ითვალისწინებენ მათში რამდენიმე ტიპის სიჭარბის არსებობას:

– *გამოსახულების არეების კოჰერენტულობა (ერთფეროვნება)* – გამოსახულების მეზობელ პიქსელებში ფერის უმნიშვნელო ცვლილება; გამოსახულების ეს თვისება გამოიყენება მისი რღვევადი კოდირებისას ყველა ცნობილი მეთოდების გამოყენებით;

– *სიჭარბე ფერით სიბრტყეებში*, რომელიც ასახავს გამოსახულების ფერითი კომპონენტების კავშირის (კორელაციის) მაღალ ხარისხს და მისი სიკაშკაშის კომპონენტის განსაკუთრებულობას;

– *მეზობელი კადრების მსგავსება* – იმ ფაქტის გამოყენება, რომ წამში 25 კადრის შემთხვევაში (ეს კი ის მინიმალური სიხშირეა, რომლის დროსაც შეუმჩნეველია კადრების ცვლილებით გამოწვეული ციმციმი) კადრებს შორის განსხვავება უმნიშვნელოა.

გასული საუკუნის 80-იანი წლებიდან მრავალი დასავლური უნივერსიტეტი და ლაბორატორია მუშაობდა ციფრული ვიდეოსიგნალის კომპრესიის ალგორითმის შექმნაზე. შეიქმნა შიდასაფირმო სტანდარტების საკმაოდ დიდი რაოდენობა. კვლევის ეს სფერო ძალზე სპეციფიური და დინამიურია, რის გამოც საერთაშორისო სტანდარტები იქმნება ალგორითმის დამუშავებიდან სულ რაღაც 2–3 წლის შემდეგ. განვიხილოთ ციფრული ვიდეოს სფეროში არსებული სტანდარტები.

1988 წელს სტანდარტიზაციის საერთაშორისო ორგანიზაციის (ISO) ფარგლებში მუშაობა დაიწყო ციფრული ვიდეოს სფეროს ექსპერტთა **MPEG (Moving Pictures Experts Group)** ჯგუფმა. 1990 წლის სექტემბერში ამ ჯგუფმა წარმოადგინა კოდირების წინასწარი სტანდარტი **MPEG-1**, რომელზეც მუშაობა დამთავრდა 1992 წელს.

აღნიშნული სამუშაოები დაიწყო არა ცარიელ ადგილზე და **MPEG**-ის, როგორც ალგორითმის, ყავს მრავალი წინამორბედი ალგორითმი. ეს, უპირველეს ყოვლისა, **JPEG**-ია, რომელიც სტატისტიკური (უძრავი) ფერადი გამოსახულებების შეკუმშვისათვის გათვალისწინებული უნივერსალური ალგორითმია. მისი უნივერსალურობა იმას ნიშნავს, რომ იგი გამოსახულებათა ფართო კლასისათვის იძლევა არც თუ ცუდ შედეგებს. ალგორითმი იყენებს რამდენიმე გარდასახვისაგან შედგენილ კონვეიერს, რომელთა შორის უმთავრესია დისკრეტული კოსინუსური გარდასახვა (**დკბ**) და რომელიც

გამოსახულების ხარისხის შესამჩნევი დანაკარგის გარეშე იძლევა შეკუმშვის კოეფიციენტის ფართო დიაპაზონში ცვლილების საშუალებას. ფრაზა "შესამჩნევი დანაკარგის გარეშე" გულისხმობს იმ ფაქტს, რომ თვალისთვის პრაქტიკულად შეუმჩნეველია განსხვავება საწყის და აღდგენილ გამოსახულებებს შორის. ალგორითმის არსი იმაშია, რომ რეალურ გამოსახულებებში მისი მატრიცის კოსინუსების ორმაგ მწკრივად განშლისას მაღალსიხშირული შემდგენების ამპლიტუდები მცირეა, რაც იძლევა მათი გადაგდების ან უხეშად წარმოდგენის შესაძლებლობას ისე, რომ იგი არ აისახოს გამოსახულების ხარისხზე. გარდა ამისა, გამოიყენება სხვა ფერით სივრცეში (YUV) გადატანა, ჯგუფური კოდირება და ხაფმენის კოდირება.

კომპრესიის ალგორითმი

MPEG–ში ვიდეოს შეკუმშვის ტექნოლოგია ორი ნაწილისაგან შედგება:

1. ვიდეოინფორმაციის სიჭარბის შემცირება დროით განზომილებაში, რომელიც ეყრდნობა იმას, რომ მეზობელი კადრები, როგორც წესი, ერთმანეთისაგან ძლიერ არ განსხვავდებიან;

2. ცალკეული გამოსახულებების შეკუმშვა.

სიჭარბის შემცირება დროით განზომილებაში

ერთმანეთის გამომრიცხავი მოთხოვნების დაკმაყოფილებისა და ალგორითმის მოქნილობის გაზრდის მიზნით მოძრავი გამოსახულების შემქმნელი უძრავი კადრების მიმდევრობიდან გამოყოფენ კადრების ოთხ ტიპს:

- *I-კადრები* – დამოუკიდებლად შეკუმშული (*I-Intraframes*);
- *P-კადრები* – ერთ გამოსახულებაზე მითითების გამოყენებით შეკუმშული (*P-Predicted*);
- *B-კადრები* – ორ გამოსახულებაზე მითითების გამოყენებით შეკუმშული (*B-Bidirection*);
- *BC-კადრები* – დამოუკიდებლად შეკუმშული ხარისხის დიდი დანაკარგებით (გამოიყენება მხოლოდ სწრაფი ძებნისას).

I-კადრები, მონაცემთა ნაკადში არიან რა დეკოდერის ერთგვარი შემავალი წერტილები, უზრუნველყოფენ თავისუფალ მიღწევას ნებისმიერ კადრამდე.

P-კადრები არქივირებისას იყენებენ მითითებას ერთ **I-** ან **P-კადრზე** და ამგვარად ზრდის მთლიანი ფილმის შეკუმშვის ხარისხს.

B-კადრები, იყენებენ რა მითითებებს ორ (წინამდებარე და მომდევნო) კადრზე, უზრუნველყოფენ შეკუმშვის უდიდეს ხარისხს; თავად ისინი კი მისათითებელი კადრების როლში ვერ გამოიყენებიან.

ფილმში კადრების მიმდევრობა შეიძლება იყოს, მაგალითად, ასეთი: **I B B P B B P B B P B B I B B P**, ან **I P B P B P B I P B P B ...**

I-კადრის სისშირეს ირჩევენ თავისუფალი მიღწევის მოთხოვნილი დროიდან და ხელშეშლებიან არხში ნაკადის გადაცემის მოთხოვნილი საიმედოობიდან გამომდინარე. **P** და **B-კადრებს** შორის თანაფარდობას კი ირჩევენ საჭირო შეკუმშვის ხარისხისა და დეკოდერის სირთულის გათვალისწინებით, ვინაიდან **B-კადრის** აღდგენისათვის უნდა გექონდეს მისი როგორც წინა, ასევე მომდევნო კადრები.

რამდენიმე გამოსახულების კოდირების შემთხვევაში ერთერთი ძირითადი ცნებაა "მაკრობლოკი". მაკრობლოკი – ესაა **16x16=256** პიქსელისაგან შედგენილი **16x16** ზომის მატრიცა (გამოსახულების ზომა უნდა იყოს 16-ის ჯერადი). ასეთი ზომა შემთხვევით არაა შერჩეული – **ღპბ** მუშაობს **8x8=64** ელემენტისაგან შედგენილ **8x8** ზომის მატრიცებზე. შეკუმშვისას თითოეული მაკრობლოკი **RGB** ფერითი სივრციდან გადაიყვანება **YUV** ფერით სივრცეში. სიკაშკაშის კომპონენტის შესაბამისი **Y** მატრიცა გარდაიქმნება **ღპბ**-ის ოთხ საწყის მატრიცად, ხოლო **U** და **V** კომპონენტების შესაბამისი მატრიცები *прореживаются* ყველა წყვილ სტრიქონზე და სვეტზე, რის გამოც გარდაიქმნება **ღპბ**-ის ერთ მატრიცად.

ამრიგად, ადამიანის თვალის იმ თავისებურებაზე დაყრდნობით, რომ იგი ცალკეული წერტილის ფერს უფრო ცუდად არჩევს, ვიდრე მის სიკაშკაშეს, ჩვენ უცებ ვღებულობთ ორჯერად შეკუმშვას.

ცალკეული მაკრობლოკები იკუმშება დამოუკიდებლად, ე.ი. **B-კადრებში** მაკრობლოკი შეიძლება შეგკუმშოთ როგორც **I-ბლოკი**, **P-ბლოკი** წინა კადრის მითითებით, **P-ბლოკი** მომდევნო კადრის მითითებით და, ბოლოს, როგორც **B-ბლოკი**.

ცალკეული კადრების კომპრესია

სტატიკური (უძრავი) გამოსახულებების შეკუმშვისათვის არსებობს მრავალი ალგორითმი. მათგან ყველაზე ხშირად გამოიყენება ალგორითმები დისკრეტული კოსინუსური გარდასახვის (**დკბ**) ბაზაზე. **MPEG**-ში ცალკეული კადრების შეკუმშვის ალგორითმი მსგავსია სტატიკური გამოსახულებების შესაბამისი **JPEG** ალგორითმისა. გავიხსენოთ, თუ როგორ გამოიყურება **JPEG**-კოდირების პროცედურა.

იმ მაკრობლოკებისადმი, რომლებიც ამზადებენ სიჭარბის შემცირების ალგორითმს დროით განზომილებაში, გამოიყენება **დკბ**. თავად გარდასახვა მდგომარეობს ორი ცვლადის ფუნქციის ორმაგ მწკრივად განშლაში გარკვეული სისშირეების კოსინუსების მიხედვით. დისკრეტული კოსინუსური გარდასახვას სიკაშკაშის მნიშვნელობათა მატრიცა გადაყავს სპექტრული კომპონენტების ამპლიტუდების მატრიცაში, ამასთან დაბალი სისშირეების შესაბამისი ამპლიტუდები ჩაიწერება მატრიცის ზედა მარცხენა კუთხეში, ხოლო შედარებით მაღალი სისშირეების შესაბამისი ამპლიტუდები – ქვედა მარჯვენაში. ვინაიდან რეალურ გამოსახულებებში მაღალსისშირული შემდგენების ამპლიტუდა ძალიან მცირეა, საბოლოოდ მიღებული მატრიცის დიაგონალის ქვემოთ არსებული მნიშვნელობები ან ნულია, ან ნულთან ახლოსაა.

ამპლიტუდების მიღებულ მატრიცაში გამოიყენება დაკვანტვის ოპერაცია. სწორედ დაკვანტვის (ჯგუფური კოდირების) ეტაპზე ხდება ძირითადად ადაპტური შეკუმშვა და აქვე წარმოიქმნება ფილმის ხარისხის ძირითადი დანაკარგები. დაკვანტვა წარმოადგენს ამპლიტუდების მატრიცის მთელ რიცხვზე ელემენტურ გაყოფას დაკვანტვის მატრიცაზე. ამ უკანასკნელის მნიშვნელობათა შერჩევა იძლევა გარკვეული სისშირეების მიხედვით დანაკარგების გაზრდის ან შემცირებისა და შესაბამისად გამოსახულების ხარისხისა და შეკუმშვის კოეფიციენტის რეგულირების საშუალებას. შევნიშნოთ, რომ გამოსახულების სხვადასხვა კომპონენტებისათვის შესაძლოა გამოყენებული იყოს დაკვანტვის საკუთარი მატრიცა.

ალგორითმის შემდეგი ეტაპი მდგომარეობს მიღებული **8x8** ზომის მატრიცის გარდაქმნაში **64**-ელემენტიან ვექტორად. ამ ეტაპს ზიგზაგ-სკანირებას უწოდებენ, ვინაიდან მატრიცის ელემენტები დაწყებული ზედა მარცხენა კუთხიდან ლაგდება ზიგზაგისებურად იმ დიაგონალების მიხედვით, რომლებიც პარალელურია ძირითადი დიაგონალისა (იხ. ნახ. 2.10). ამის შედეგად მიიღება

ვექტორი, რომლის საწყის პოზიციებზე იმყოფება დაბალი სიხშირეების შესაბამისი მატრიცის ელემენტები (კომპონენტები), ხოლო უკანასკნელ პოზიციებზე – მაღალსიხშირული კომპონენტები. ამრიგად, ვექტორის ბოლოს იქნება ნულოვანი კომპონენტების ძალიან დიდი რაოდენობა.

ზემოაღწერილის შემდეგ მეორდება ყველა მოქმედებები, რომლებიც შეესაბამებოდა უძრავი (სტატიკური) გამოსახულებების შეკუმშვის სტანდარტულ **JPEG** ალგორითმს.

ბლოკების წანაცვლების (გადაადგილების) ვექტორების გამოყენება

მეზობელი კადრების მსგავსობის გათვალისწინების უმარტივესი ხერხი იქნებოდა წინმსწრები (წინამდებარე) კადრის ყველა ბლოკიდან მიმდინარე კადრის შესაბამისი ბლოკის გამოკლება. თუმცა გაცილებით ეფექტურია იმ ვექტორების სიგრძის ძებნის (პოვნის) ალგორითმი, რომლებზედაც აღმოჩნდა დაძრული მიმდინარე კადრის ბლოკები წინამდებარე კადრთან შედარებით.

ალგორითმი იმაში მდგომარეობს, რომ გამოსახულების მიმდინარე კადრის ყოველი ბლოკისათვის ჩვენ წინამდებარე კადრის შესაბამის მიდამოში (არეში) ვპოულობთ ბლოკს, რომელიც ახლოსაა მასთან გარკვეული პარამეტრის (მაგალითად, პიქსელების სხვაობების კვადრატების ჯამის მინიმუმის) მიხედვით. თუ ბლოკებს შორის მინიმალური მანძილი ნაკლებია აღმოჩნდება რაიმე ზღურბლურ სიდიდეზე, მაშინ გამომაგალ ნაკადში ყოველ ბლოკთან ერთად შენარჩუნდება წანაცვლების (გადაადგილების) ვექტორი, ანუ წინა ***I*** ან ***-კადრში*** მაქსიმალურად მსგავსი ბლოკის კოორდინატები. იმ შემთხვევაში, როდესაც განსხვავება აღემატება ზღურბლს, მაშინ მიმდინარე ბლოკი იკუმშება დამოუკიდებლად.

2.10. მატრიცების სიბნალების კომპრესიის მეთოდები

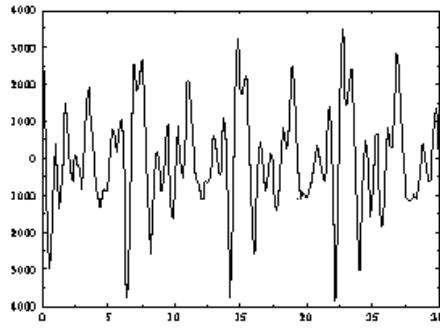
რაც არ უნდა ითქვას, ადამიანებს სორის ურთიერთობის ძირითადი ხერხი იყო და არის ლაპარაკი და და სალაპარაკო შეტყობინებების გადაცემა. კავშირის სისტემებში გადაცემული ინფორმაციის ძირითადი ნაწილი დღესაც უჭირავს სიტყვას (სამავთულო ტელეფონია, ფიჭური და თანამგზავრული კავშირის სისტემები, და ა.შ.). ამიტომ შესაბამისი სიგნალების ეფექტურ კოდირებას, ანუ შეკუმშვას ექცევა გამორჩეული ყურადღება.

მეტყველების სიგნალების შეკუმშვის ისტორია ითვლის რამდენიმე ათეულ წელს. მაგალითად, იმ დროში, როდესაც შეკვეთილი სატელეფონო საუბრისათვის ლოდინის დრო შეადგენდა ათეულობით საათს, ეკონომიურმა შეზღუდვებმა გამოიწვია ეგრეთ წოდებული **J2** აპარატურის დაყენება აშშ-ის ტრანსკონტინენტალურ ხაზებსა და ატლანტიკის ოკეანეში ჩადებულ კაბელზე, რომელთა არხების ზოლი შეადგენდა **0,3 – 3,5** კმ-ს. ასეთი აპარატურა ოდესაღაც მუშაობდა აგრეთვე მოსკოვი–ვლადივოსტოკის ხაზზე. მისი არხების ხარისხი ძლივს აღწევდა ორბალიან ნიშნულს, მაგრამ, მიუხედავად ამისა, გადამწყვეტი აღმოჩნდა სატელეფონო შეერთებების ორჯერადი გაზრდა. იმ დროს არხებზე მომხმარებელთა მოთხოვნებმა კავშირის ხარისხი გადააქცია მეორეხარისხოვან ფაქტორად. დღეს კი ხარისხის ფაქტორი არანაკლებ მნიშვნელოვანია, ვიდრე კავშირის არხების გამტარუნარიანობის ეკონომია.

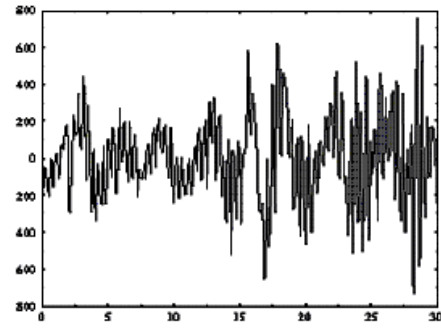
განვიხილოთ მეტყველების სიგნალის, როგორც ეკონომიური კოდირებისა და კავშირის ხაზებით გადაცემის ობიექტის, ძირითადი თავისებურებები და შევეცადოთ იმის გარკვევას, თუ სიგნალის რომელ თვისებებს ემყარება მისი შეკუმშვა (კომპრესია).

სალაპარაკო შეტყობინება წარმოადგენს რთული ფორმის რხევას, რომელიც დამოკიდებულია წარმოთქმულ სიტყვებზე, ხმის ტემპზე, ინტინაციასა და მოსაუბრის სქესსა და ასაკზე. ლაპარაკის სპექტრი საკმაოდ განიერია და შეადგენს დაახლოებით 50-დან 10000 ჰც-დე ღიაპაზონს, თუმცა ანალოგურ ტელეფონიაში მისი გადაცემისათვის ოდესღაც უარი თქვეს 0,3...3,4 კმ ღიაპაზონზე გარეთ მყოფ სიხშირულ შემდგენებზე. ამან რამდენადმე გააუარესა მთელი რიგი ბგერების აღქმა (მაგალითად, მოშიშინე ბგერები, რომელთა ენერჯის მნიშვნელოვანი ნაწილი თავმოყრილია 0,3...3,4 კმ ღიაპაზონის გარეთ), თუმცა გარჩევადობა თითქმის შენარჩუნდა. სიხშირის ქვემოდან შეზღუდვა (300 ჰც-დე) ასევე რამდენადმე აუარესებს აღქმას ძირითადი ტონის დაბალსიხშირული შემდგენების დაკარგვის გამო.

წარმოდგენილ ნახაზებზე გამოსახულია მეტყველების სიგნალების ფრაგმენტები, რომლებიც შეესაბამებიან *ხმოვან* (ნახ. 2.16) და *თანხმოვან* (ნახ. 2.17) ბგერებს, და აგრეთვე ამ სიგნალების სპექტრები (ნახ. 2.18 და 2.19).

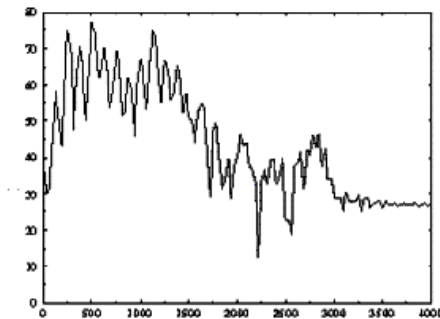


ნახ. 2.16. ხმოვანი ბგერებისაგან
ბგერებისაგან
შედგენილი სიგნალის ფრაგმენტი



ნახ. 2.17. თანხმოვანი
შედგენილი სიგნალის ფრაგმენტი

ნახაზებიდან კარგად ჩანს შესაბამისი სიგნალების მახასიათებლებს სორის განსხვავება და, აგრეთვე, ის, რომ როგორც პირველ, ასევე მეორე შემთხვევაში სიგნალის სპექტრის სიგანე არ აღემატება 3,5 კჰც-ს. გარდა ამისა, შეიძლება აღნიშნოს, რომ მეტყველების სიგნალის სპექტრში დაბალსიხშირული (ანუ დროში მდორედ ცვლადი) შემდგენების დონე გაცილებით აღემატება მაღალსიხშირული (სწრაფი) შემდგენების დონეს. სპექტრის ეს არსებითი არათანაბრობა წარმოადგენს ასეთი სიგნალების კუმშვადობის ერთერთ ფაქტორს.



ნახ. 2.18.

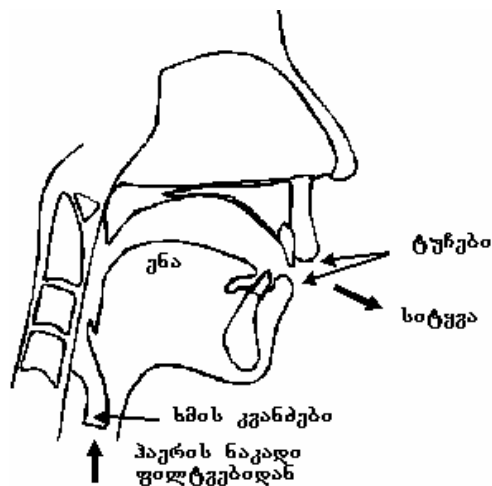


ნახ. 2.19

მეტყველების სიგნალების მეორე თავისებურებას, როგორც ეს მოყვანილი მაგალითებიდან ჩანს, წარმოადგენს სიგნალის მყისა მნიშვნელობების ალბათობათა განაწილების (ალბათობის სიმკვრივის) არათანაბრობა. კერძოდ, სიგნალის მცირე დონეები გაცილებით უფრო ალბათურია, ვიდრე დიდი დონეები. ეს განსაკუთრებით შეიმჩნევა ისეთ ხანგრძლივ ფრაგმენტებზე, რომლებიც შეესაბამება მეტყველების დაბალ აქტივობას. აღნიშნული ფაქტორი, როგორც ცნობილია, ასევე იძლევა ეკონომიური კოდირების შესაძლებლობას – უფრო ალბათური მნიშვნელობები შეიძლება კოდირებული იყოს მოკლე კოდებით, ხოლო ნაკლებად ალბათური – გრძელი კოდებით.

მეტყველების სიგნალების კიდევ ერთი თავისებურებაა მათი მნიშვნელოვანი არასტაციონალურობა დროში: სიგნალის თვისებები და პარამეტრები სხვადასხვა უბნებზე მნიშვნელოვნად განსხვავდებიან ერთმანეთისაგან. ამასთანავე სტაციონალურობის ინტერვალის ზომა რამდენიმე ათეულ მილიწამის რიგისაა. სიგნალის ეს თვისება ძალიან ართულებს მის ეკონომიურ კოდირებას, ვინაიდან მოითხოვს შეკუმშვის სისტემის ადაპტურობას. ეს კი იმას ნიშნავს, რომ სიგნალის ყოველ უბანზე სისტემა უნდა ეწყობოდეს სიგნალის პარამეტრებს.

და ბოლოს, მეტყველების სიგნალების შეკუმშვის ორგანიზებისათვის განსაკუთრებული მნიშვნელობა აქვს ლაპარაკის წარმოქმნის მექანიზმის ფიზიკის ცოდნას. მისი გამარტივებული სქემა ნაჩვენებია 2.20 ნახაზზე.



ნახ. 2.20. მეტყველების წარმოქმნის მექანიზმის გამარტივებული სქემა

სიტყვა ფორმირდება ფილტვებიდან ამოსული ჰაერის ნაკადის ხმის კვანძებსა და ხმის ტრაქტში გავლის შედეგად. ხმის ტრაქტი იწყება ხმის კვანძებიდან მისი სიგრძე საშუალოდ 15...17 სმ-ია. ხმის ტრაქტს, თავისი რეზონანსული თვისებების გამო, საფორმირებელ სიგნალში შეაქვს ყოველი ადამიანისათვის დამახასიათებელი სიხშირული შემდგენების ნაკრები, რომლებსაც *ფორმანტებს* უწოდებენ. მათი სიხშირეები და ზოლები შეიძლება იმართებოდეს ხმის ტრაქტის ფორმის ცვლილებით, მაგალითად, ენის მდებარეობის შეცვლით. მეტყველების მრავალი კოდერისა და დეკოდერის უმთავრესი ნაწილია ხმის ტრაქტის მოდელირება ცვალებადი (რეგულირებადი) პარამეტრებიანი ფილტრის სახით. ვინაიდან ხმის ტრაქტის ფორმა შეიძლება იცვლებოდეს შედარებით მდორედ (ძნელი წარმოსადგენია, რომ ენის მდებარეობა იცვლებოდეს უფრო ხშირად, ვიდრე 20...30-ჯერ 1 წმ-ის განმავლობაში), ამიტომ

ასევე შედარებით იშვიათად უნდა ახლდებოდეს (ან იცვლებოდეს) ასეთი ფილტრის პარამეტრები (ჩვეუბრივ ყოველი 20 მწმ-ის შემდეგ ან უფრო იშვიათად).

ამრიგად, ხმის ტრაქტი აღიგზნება ხმის კვანძების გავლით მისკენ მიმართული ჰაერით. ამ დროს წარმოქმნილი ბგერები მათი აღგზნების ხერხისაგან დამოკიდებულებით შეიძლება დავეოთ სამ კლასად:

1. ხმოვანი ბგერები, რომლებიც წარმოიქმნება ხმის კვანძების ვიბრაციის შედეგად, რომელთა ჩაკეტვა და გაღება იწვევს ფილტვებიდან ხმის ტრაქტისაკენ მომავალი ჰაერის ნაკადის შეწყვეტას ან მიწოდებას. ამასთან ხმის ტრაქტის აღგზნება ხდება კვაზიპერიოდული იმპულსებით. კვანძების გაღებისა და ჩაკეტვის სიჩქარე (სიხშირე) განსაზღვრავს წარმოქმნილ ხმამაღლობას (ტონს). მისი რეგულირება (მართვა) შესაძლებელია *ხმის კვანძების ფორმისა და დაძაბულობის და, აგრეთვე, ჰაერის მიწოდებული ნაკადის წნევის ცვლილებით*.

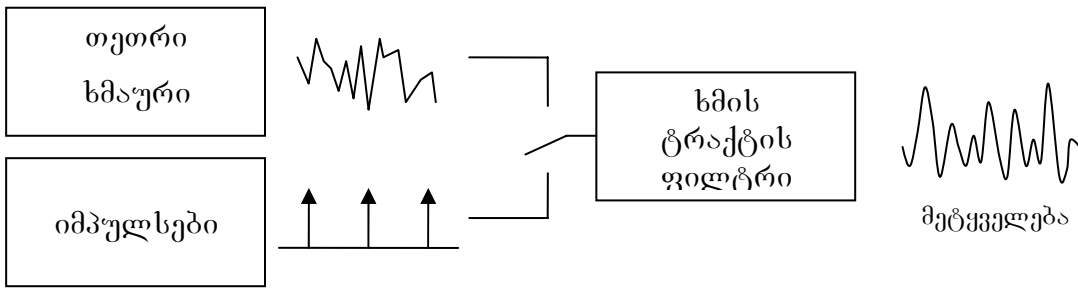
ხმოვან ბგერებს გააჩნიათ 2...20 მწმ-ის ტოლი პერიოდის მქონე ძირითადი ტონის პერიოდულობის მაღალი ხარისხი. იგი კარგად ჩანს 2.16 ნახაზზე, სადაც მოყვანილია ხმოვანი ბგერებისაგან შედგენილი მეტყველების სიგნალის ფრაგმენტი.

2. თანხმოვანი ბგერები წარმოიქმნება ხმაურის მსგავსი ტურბულენტური ნაკადის აღძვრისას, რომელიც ფორმირდება ხმის ღია კვანძების გავლით დიდი სიჩქარის ჰაერის ნაკადის გავლის შედეგად. ასეთ ბგერებში, როგორც 2.17 ნახაზიდან ჩანს, პრაქტიკულად არ არსებობს ხმის კვანძების ვიბრაციით გამოწვეული ხანგრძლივი პერიოდულობა, თუმცა აღინიშნება ხმოვანი ტრაქტის გავლენით გამოწვეული მცირე ხანგრძლივობის კორელაცია.

3. ფეთქებადი ხასიათის ხმები, რომლებიც წარმოიქმნება მაშინ, როდესაც ხმის ჩაკეტილი ტრაქტი უცაბედად გაიხსნება ჰაერის მოჭარბებული წნევის მოქმედების შედეგად.

ზოგიერთი ხმა სუფთად არ მიესადაგება არცერთ ზემოაღწერილ კლასს, მაგრამ ისინი შეიძლება განვიხილოთ როგორც მათი ნაზავი (ერთობლიობა).

ამრიგად, მეტყველების წარმოქმნის პროცესი შეიძლება განვიხილოთ როგორც აღგზნების სიგნალის გაფილტვრა ხმის წარმოქმნელი ტრაქტის მიერ, რომელსაც გააჩნია როგორც დროში ცვლადი პარამეტრები, ასევე ცვლადი მახასიათებლები (ნახ. 2.21).



ნახ. 2.21. მეტყველების წარმოქმნის პროცესის ილუსტრაცია

ამასთანავე უნდა აღინიშნოს, რომ, მეტყველების გენერირებული (ფორმირებული) სიგნალების მრავალფეროვნობის მიუხედავად, *მეტყველების (სალაპარაკო) ტრაქტის ფორმა და პარამეტრები და, აგრეთვე, ალგზნების ხერხები და პარამეტრები საკმაოდ ერთგვაროვანია და ისინი იცვლებიან შედარებით მდორედ.* 2.16 და 2.17 ნახაზებიდან კარგად ჩანს, რომ მეტყველების სიგნალს გააჩნია მოკლევადიანი და ხანგრძლივადიანი წინასწარმეტყველობის მაღალი ხარისხი, რაც განპირობებულია ხმის კვანძების ვიბრაციის პერიოდულობითა და ხმის ტრაქტის რეზონანსული ხასიათით. *ხმის კოდერებისა და დეკოდერების უმრავლესობა კოდის სიჩქარის შემცირებისათვის იყენებს წინასწარმეტყველების აღნიშნულ შესაძლებლობას და, აგრეთვე, ხმის წარმოქმნელი სისტემის მოდელის პარამეტრების მდორედ ცვლილებას.*

უნდა აღინიშნოს, რომ მეტყველების სიგნალების ეკონომიური კოდირების ყველა ცნობილი მეთოდი პირობითად შეიძლება დავეყოსთ ქვემოთ აღწერილ სამ კლასად.

2.10.1. სიბნალის ფორმის კოდირება

მეტყველების კოდერებიდან და დეკოდერებიდან უმარტივესია *სიბნალის ფორმის კოდერები და დეკოდერები*, რომლებიც სრულიადაც არ იყენებენ ინფორმაციას იმის შესახებ, თუ როგორ ფორმირდა სიბნალი, არამედ უბრალოდ ცდილობენ აღდგენილი სიბნალის ფორმა მაქსიმალურად მიუახლოვონ ორიგინალს. თეორიულად ისინი მათ შესასვლელზე მიწოდებული სიბნალის ხასიათის მიმართაა ინვარიანტულები და შეიძლება გამოყენებულნი იყვნენ ნებისმიერი, მათ შორის არამეტყველების, სიბნალის კოდირებისათვის. ეს კოდერები უმარტივესია მოქმედების პრინციპისა და მოწყობილობის

თვალსაზრისით, მაგრამ ისინი ვერ უზრუნველყოფენ შეკუმშვის მაღალ ხარისხს (კოდის დაბალ სიჩქარეებს).

სიგნალის ფორმის კოდირების უმარტივესი მეთოდია *იმპულსურ-კოდური მოდულაცია* – *ოპმ* (ანუ *PCM* – Pulse Code Modulation), რომლის გამოყენებისას ხდება შემავალი სიგნალის უბრალოდ დისკრეტიზაცია, თანაბარი დაკვანტვა და მიღებული შედეგის გარდაქმნა თანაბარ ორობით კოდად.

მეტყველების სიგნალებისათვის მათი 0,3...3,4 კჰც სტანდარტულ ზოლში გადაცემისას ჩვეულებრივ იყენებენ დისკრეტიზაციის სიხშირეს, რომელიც $F_{დისკრ.} \geq 2F_{გაქს.} = 8$ კჰც-ის ტოლია. ექსპერიმენტულად ნაჩვენებია, რომ თანაბარი დაკვანტვისას მეტყველების თითქმის იდეალური ხარისხის მისაღებად სიგნალი უნდა დაიკვანტოს არა ნაკლებ ± 2000 დონით, ანუ ყოველი ანათვალის წარმოსადგენად საჭირო იქნება **12 ბიტი**, რის გამოც კოდის ჯამური სიჩქარე იქნება

$$R = 8000 \text{ (ანათვალი/წმ)} \cdot 12 \text{ (ბიტი/ანათვალი)} = 96000 \text{ ბიტი/წმ} = 96 \text{ კბიტი/წმ.}$$

არათანაბარი დაკვანტვის გამოყენებისას (ანუ სიგნალის დაბალი დონეებისათვის უფრო ზუსტი და მაღალი დონეებისათვის უფრო უხეში სკალის გამოყენებას ისე, რომ დაკვანტვის ფარდობითი (შედარებითი) შეცდომა ყველა დონისათვის მუდმივი (ერთნაირი) აღმოჩნდეს), შეიძლება მივაღწიოთ მეტყველების აღდგენილი სიგნალის ისეთივე სუბიექტურ ხარისხს, როგორც მიიღებოდა თანაბარი დაკვანტვისას, მაგრამ არათანაბარი დაკვანტვის შემთხვევაში დაკვანტვის დონეთა რაოდენობა აღმოჩნდება გაცილებით ნაკლები, კერძოდ კი ± 128 . ამ დროს სიგნალის ანათვლების ორობითი წარმოდგენისათვის საკმარისი იქნება **8 ბიტი** და კოდის საბოლოო სიჩქარე შეადგენს **64 კბიტი/წმ-ს**.

მეტყველების სიგნალების სტატისტიკური (ანუ მათი მყისა მნიშვნელობების ალბათობების განაწილების სახის) და სმენის არაწრფივი თვისებების (რის გამოც ადამიანი გაცილებით უკეთ არჩევს სუსტ ხმებს) გათვალისწინებით ოპტიმალურია დაკვანტვის ლოგარითმული სკალა, რომელიც გასული საუკუნის 60-იან წლებში მიიღეს როგორც სტანდარტი და იგი დღესაც გამოიყენება. თუმცა არაწრფივი დაკვანტვის ევროპული და აშშ-ის სტანდარტები რამდენადმე განსხვავდება ერთმანეთისაგან (**μ -law companding** и **A-law compression**), რაც იწვევს სიგნალების ხელახალი კოდირების აუცილებლობას.

ამრიგად, მეტყველების სიგნალების ეფექტურობისა და კოდირების ხარისხის ნებისმიერი შედარებისათვის საწყისად გამოდგება კოდის სიჩქარე, რომელიც **64 კბიტ/წმ**-ის ტოლია.

კოდის ჯამური სიჩქარის შემცირების რეალიზაციისათვის შემდეგი ხერხი შეიძლება იყოს სიგნალის მიმდინარე ანათვლის წინასწარმეტყველების მცდელობა რამდენიმე წინმსწრები ანათვლის მიხედვით და, შემდეგ, კოდირება არა მიმდინარე ანათვლისა, არამედ წინასწარმეტყველების შეცდომისა, რომელიც *წარმოადგენს სხვაობას მიმდინარე ანათვლის ნამდვილ და მის ნაწინასწარმეტყველევ მნიშვნელობებს შორის*. თუ წინასწარმეტყველების სიზუსტე საკმაოდ მაღალია, მაშინ მომდევნო ანათვლის წინასწარმეტყველების შეცდომა იქნება მნიშვნელოვნად ნაკლები თვით ანათვლის მნიშვნელობასთან შედარებით და, ამიტომ, მისი კოდირებისათვის საჭირო იქნება ბიტების გაცილებით მცირე რაოდენობა. ამრიგად, *რაც უფრო მეტად იქნება შესაძლებელი სიგნალის ქცევის წინასწარმეტყველება, მით უფრო ეფექტური იქნება მისი შეკუმშვა (კომპრესია)*.

ზემოაღწერილი იდეა უდევს საფუძვლად ეგრეთ წოდებულ დიფერენციალურ იმპულსურ-კოდურ მოდულაციას – **დიკმ-ს (DPCM)** – კოდირების მეთოდს, რომლის დროსაც ხდება არა სიგნალის ანათვლების, არამედ *გარკვეული სახით ფორმირებულ ნაწინასწარმეტყველევ მნიშვნელობებს შორის განსხვავებების კოდირება*. წინასწარმეტყველების უმარტივესი ხერხია მიმდინარე ანათვლის წინასწარმეტყველებისათვის სიგნალის წინა ანათვლის გამოყენება:

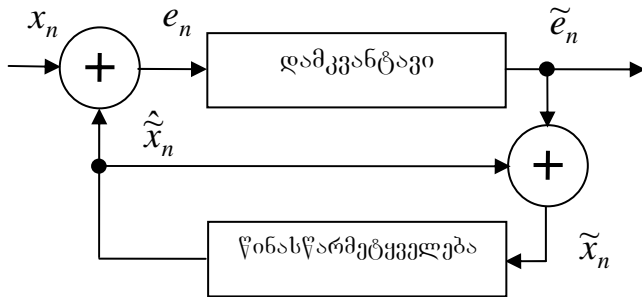
$$x^*_i = x_{i-1}, \quad \varepsilon_i = x_i - x^*_i. \quad (2.22)$$

ეს ეგრეთ წოდებული *ნულოვანი რიგის წინასწარმეტყველება*ა, რომელიც ყველაზე მარტივი და, ამავე დროს, ყველაზე არაზუსტია. ცხადია, რომ უფრო ზუსტია მიმდინარე ანათვლის წინასწარმეტყველება წინა ორი და ა.შ. ანათვლის წრფივი კომბინაციის საფუძველზე:

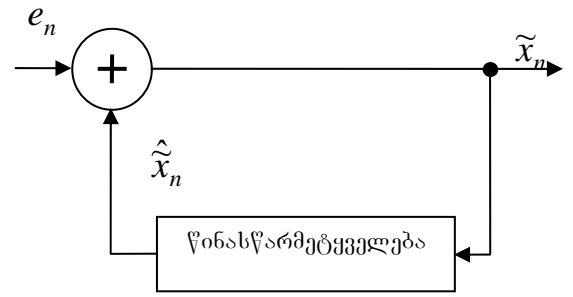
$$x^*_i = \sum a_k x_{i-k}, \quad \varepsilon_i = x_i - x^*_i. \quad (2.23)$$

სამწუხაროდ, წინასწარმეტყველების რიგის ზრდის კვალობაზე წინასწარმეტყველების სიზუსტე ყოველთვის როდი იზრდება, ვინაიდან სიგნალის თვისებები ანათვლებს შორის უკვე იწყებს ცვლილებებს, რის გამოც ჩვეულებრივ იყენებენ მე-2 ან მე-3 რიგის წინასწარმეტყველებას

2.22 და 2.23 ნახაზებზე წარმოდგენილია **დიკმ (DPCM)** კოდერისა და დეკოდერის სქემები.



ნახ. 2.22. **დიკმ** კოდერის სქემა



ნახ. 2.23. **დიკმ** დეკოდერის სქემა

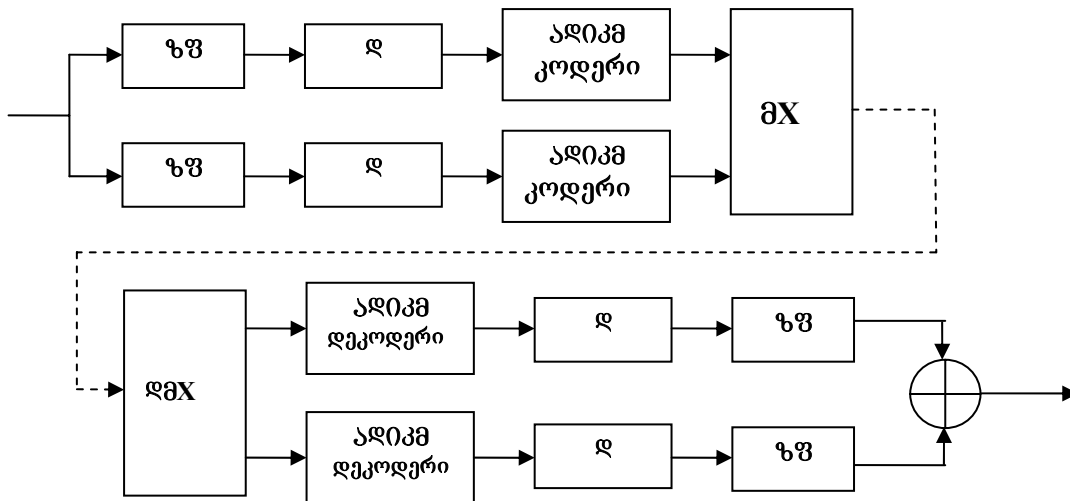
მეტყველების სიგნალების კოდირებისას მათი მცირეხანგრძლივობიანი (მხოლოდ რამდენიმე ანათვალზე) წინასწარმეტყველების შესაძლებლობის გათვალისწინებით კოდის ერთ ანათვალზე გადათვლილი (გადაანგარიშებული) საბოლოო სიჩქარე **დიკმ**-სათვის ჩვეულებრივ შეადგენს **5...6 ბიტს**, ანუ **40...48 კბიტი/წმ-ს**.

დიკმ-ის ეფექტურობა რამდენადმე შეიძლება ამაღლდეს, თუ სიგნალის წინასწარმეტყველება და დაკვანტვა განხორციელდება არა მის გასაშუალებულ მახასიათებლებზე, არამედ მათი მიმდინარე მნიშვნელობისა და ამ მნიშვნელობების დროში ცვლილების გათვალისწინებით, ანუ ადაპტურად. ასე მაგალითად, თუ სიგნალის ცვლილების სიჩქარე გაიზრდება, მაშინ შეიძლება დაკვანტვის ბიჯის გაზრდა და, პირიქით, თუ სიგნალმა ნელა (მდორედ) დაიწყო ცვლილება, მაშინ შეიძლება შევამციროთ დაკვანტვის ბიჯი. ამასთანავე წინასწარმეტყველების სეცლომა შემცირდება და, შედეგად, მისი კოდირება შესაძლებელი იქნება ერთ ანათვალზე გადაანგარიშებული ბიტების ნაკლები რაოდენობით. კოდირების ასეთ ხერხს **ადაპტურ დიკმ-ს**, ანუ **ადიკმ-ს (ADPCM)**. კოდირების ეს მეთოდი უკვე სტანდარტიზირებულია და იგი ფართოდ გამოიყენება მეტყველების შეკუმშვისათვის კავშირის საქალაქთაშორისო ციფრულ სისტემებში, კავშირის მიკროფიჭურ **DECT** სისტემაში, ციფრულ უსადენო ტელეფონიაში და ა.შ. **4 ბიტი/ანათვალზე**, ანუ **32 კბიტი/წმ** სიჩქარიანი **ადიკმ**-ის გამოყენება უზრუნველყოფს მეტყველების ისეთივე სუბიექტურ ხარისხს, როგორცაა **64 კბიტი/წმ**-იანი **იკმ**-ის შემთხვევაში, ოღონდ **ადიკმ**-ისას კოდის სიჩქარე ორჯერაა შემცირებული **იკმ**-თან შედარებით.

დღეისათვის სტანდარტიზირებულია აგრეთვე **40, 24 და 16 კბიტი/წმ**-იანი **პლიტ**-ები (უკანასკნელ შემთხვევაში სიგნალის ხარისხი რამდენადმე გაუარესებულია **32 კბიტი/წმ**-იან **პლიტ**-თან შედარებით). ამრიგად, ჩანს, რომ მეტყველების სიგნალების შეკუმშვა მათი ფორმის კოდირების საფუძველზე უკეთეს შემთხვევაში უზრუნველყოფს კოდის სიჩქარის ორჯერად ან სამჯერად შემცირებას. სიჩქარის შემდგომი შემცირება იწვევს კოდირებული სიგნალის ხარისხის მკვეთრ გაუარესებას.

სიგნალის ფორმის ზემოთ აღწერილი კოდერები იყენებენ ამ სიგნალის აღწერის წმინდა დროით მიდგომას. თუმცა შესაძლებელია სხვა მიდგომების გამოყენებაც, რომელთა მაგალითია ეგრეთ წოდებული *ქვედიაპაზონების კოდირება (Sub-Band Coding - SBC)*, რომლის დროსაც შემავალი სიგნალი დაიყოფა (ან იფილტრება) რამდენიმე სიხშირულ დიაპაზონად (*ქვედიაპაზონებად - sub-bands*) და თითოეული ქვედიაპაზონის სიგნალის კოდირება ხდება ცალ-ცალკე, მაგალითად, **პლიტ**-ის ტექნიკის გამოყენებით.

ვინაიდან თითოეულ ქვედიაპაზონს გააჩნია უფრო ვიწრო ზოლი (ყველა ქვედიაპაზონი ჯამში იძლევა საწყისი სიგნალის ზოლს), ამიტომ თითოეული მათგანის დისკრეტიზაციის სიხშირე იქნება ნაკლები. ამის შედეგად ყველა ქვედიაპაზონის შესაბამისი კოდების ჯამური სიჩქარე უკიდურეს შემთხვევაში არ გადააჭარბებს საწყისი სიგნალის შესაბამისი კოდის სიჩქარეს. აღნიშნულ ტექნიკას გააჩნია გარკვეული ღირსებები. საქმე იმაშია, რომ სმენის სუბიექტური მგრძობიარობა სიგნალებისადმი და მათი დამახინჯებებისადმი სხვადასხვა სიხშირეებზე სხვადასხვაა. იგი მაქსიმალურია **1...1,5 კჰც** სიხშირეებზე და მცირდება უფრო დაბალ და უფრო მაღალ სიხშირეებზე. ამრიგად, სმენის უფრო მაღალი მგრძობიარობის დიაპაზონში სიგნალს თუ დაკკვანტავთ უფრო ზუსტად, ხოლო ნაკლებად მგრძობიარობის დიაპაზონებში უფრო უხეშად, მაშინ შეიძლება მივიღოთ მოგება კოდის საბოლოო სიჩქარეში. მართლაც, *ქვედიაპაზონების კოდი-რების* ტექნოლოგიის გამოყენებისას კოდირებული მეტყველების კარგი ხარისხი მიიღება **16...32 კბიტი/წმ** სიჩქარის შემთხვევაში. ამ დროს კოდერი რამდენადმე რთულია მარტივ **პლიტ** კოდერთან შედარებით, თუმცა გაცილებით მარტივია მეტყველების შეკუმშვის სხვა ეფექტური მეთოდების შესაბამის კოდერთან შედარებით. მსგავსი კოდერის გამარტივებული სქემა სიგნალის სიხშირული ზოლის ორ ქვედიაპაზონად დაყოფის შემთხვევაში მოყვანილია 2.24 ნახაზზე.



ნახ. 2.24. ადიკმ კოდერის გამარტივებული სქემა

ქვედიაპაზონების კოდირებასთან ახლოსაა შეკუმშვის მეთოდი, რომელიც ემყარება სიგნალის წრფივ გარდასახვებს, მაგალითად, დისკრეტულ კოსინუსურ ან სინუსურ გარდასახვებს. მეტყველების კოდირებისათვის გამოიყენება გარდასახვის კომპონენტების ადაპტური კოდირების ტექნოლოგია **ATC (Adaptive Transform Coding)**, რომლის დროსაც სიგნალი დაიყოფა ბლოკებად, თითოეული ბლოკისადმი გამოიყენება დისკრეტული კოსინუსური გარდასახვა და ამის შედეგად მიღებული კომპონენტები (კოეფიციენტები, ტრანსფორმანტები) იკვანტება ადაპტურად სიგნალის სპექტრის თავისებურებების შესაბამისად. რაც უფრო მნიშვნელოვანია გარდასახვის კოეფიციენტები, მით უფრო დიდი რაოდენობის ბიტებით ხდება მათი კოდირება. კოდირების ტექნიკა ზალიან გავს **JPEG**-ის, მაგრამ ამ შემთხვევაში გამოიყენება მეტყველების სიგნალებისათვის. ასეთი კოდირებისას სიგნალის საკმაოდ დამაკმაყოფილებელი ხარისხის შემთხვევაში კოდის მიღწეული სიჩქარეა **12...16 კბიტ/წმ**. მეტყველების შეკუმშვისათვის ეს მეთოდი ფართოდ არ გამოიყენება, ვინაიდან ცნობილია მასთან შედარებით გაცილებით უფრო ეფექტური და რეალიზაციის თვალსაზრისით მარტივი კოდირების მეთოდები.

მეტყველების სიგნალების კოდირების შემდგომი დიდი კლასია წყაროს კოდირები.

2.10.2. წყაროს კოდირება

სიგნალის ფორმის კოდირებისაგან განსხვავებით, რომლებიც საერთოდ არ იყენებენ ინფორმაციას იმის შესახებ, თუ როგორ მოხდა საკოდირებელი სიგნალის ფორმირება, წყაროს კოდირების მუშაობა ემყარებასწორედ წყაროს მოდელს და საკოდირებელი სიგნალიდან იღებენ ინფორმაციას ამ მოდელის

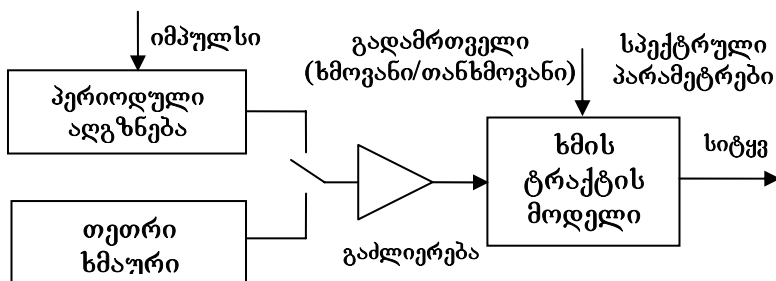
პარამეტრების შესახებ. ამასთან კოდირების შედეგია არა სიგნალების კოდები, არამედ ამ სიგნალების წყაროს პარამეტრების კოდები.

მეტყველების კოდირებისათვის განკუთვნილ წყაროს კოდერებს ვოკოდერებს (VOICE CODERS) უწოდებენ და ისინი მუშაობენ დაახლოებით შემდეგნაირად.

ხმის წარმომქმნელი ტრაქტი (ნახ. 2.25) განიხილება დროში ცვლადპარამეტრებიანი ფილტრის სახით, რომელიც აღიგზნება ან თეთრი ხმაურის წყაროს (თანხმოვანი ბგერების ფორმირებისას) ან ძირითადი ტონის პერიოდის მქონე იმპულსების მიმდევრობებით (ხმოვანი ბგერების ფორმირებისას).

ხმის წარმომქმნელი სისტემის წრფივი მოდელი და მისი პარამეტრები შეიძლება განისაზღვროს სხვადასხვა ხერხით, რაც, თავის მხრივ, განაპირობებს ვოკოდერის ტიპს.

მეტყველების სიგნალის ანალიზის შედეგად ვოკოდერის მიერ მიღებულ და დეკოდერისაკენ გადაცემულ ინფორმაციას წარმოადგენს ხმის წარმომქმნელი ფილტრის პარამეტრები, ხმოვანი/თანხმოვანი ბგერების მაჩვენებელი, აღგზნების სიგნალის სიმძლავრე და ძირითადი ტონის პერიოდი ხმოვანი ბგერებისათვის. სიგნალის არასტაციონალურობისადმი მიდევნების მიზნით ეს პარამეტრები უნდა ახლდებოდეს ყოველ 10...20 მწმ-ში.



ნახ. 2.25. ხმის წარმომქმნელი ტრაქტის მოდელი

ვოკოდერი, სიგნალის ფორმის კოდერისაგან განსხვავებით, ცდილობს აფორმროს სიგნალი, რომელიც ისე უღერს როგორც ორიგინალური სიტყვა და ყურადღებას არ აქცევს ამ სიგნალის ფორმის განსხვავებას საწყისი სიგნალის ფორმისაგან. ამასთან ვოკოდერის გამოსასვლელზე მიღებული კოდის საბოლოო სიჩქარე არ აღემატება 2,4 კბიტი/წმ-ს, ანუ თხუთმეტჯერ ნაკლებია ალიკვთან შედარებით!

სამწუხაროდ, ვოკოდერების მიერ უზრუნველყოფილი მეტყველების ხარისხი ძალზე შორსაა იდეალურისაგან, მისი ხმოვანება თუმცა საკმაოდ გარჩევადია, მაგრამ აბსოლუტურად არანატურალურია (არაბუნებრივია). ამასთანავე კოდის

სიჩქარის მნიშვნელოვანი გაზრდაც კი პრაქტიკულად არ აუმჯობესებს მეტყველების ხარისხს, ვინაიდან კოდირებისათვის შერჩეული იყო ხმის წარმომქმნელი სისტემის ძალზე მარტივი მოდელი. განსაკუთრებით უხეშია ის ვარაუდი, რომ მეტყველება შედგება მხოლოდ ხმოვანი და თანხმოვანი ბგერებისაგან, ხოლო შუალედური მდგომარეობები არ არსებობს.

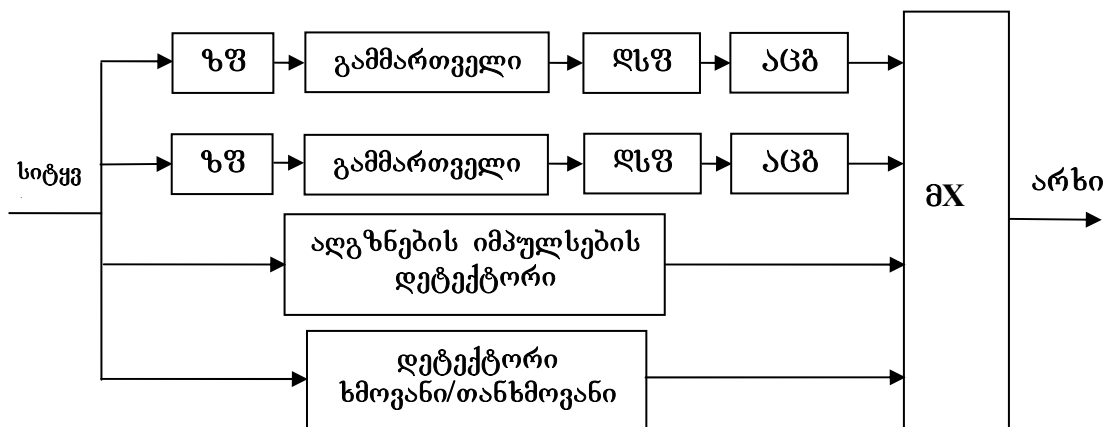
ვოკოდერები ძირითადად გამოიყენება სამხედრო საქმეში, სადაც მთავარია არა სიტყვის ნატურალურობა, არამედ მისი შეკუმშვის მაღალი ხარისხი და კოდის ძალზე მცირე სიჩქარე, რაც იძლევა გადასაცემი სიტყვის გასაიდუმლოების საშუალებას.

მოკლედ განვიხილოთ ცნობილი ვოკოდერების ძირითადი ტიპები.

საარხო ვოკოდერები

საარხო ვოკოდერის ტიპი წარმოდგენილი იყო ჯერ კიდევ 1939 წელს. იგი იყენებს ადამიანის სმენის სუსტ მგრძობიარობას სიგნალის უმნიშვნელო ფაზური (დროითი) ძვრების მიმართ.

მეტყველების სიგნალის დაახლოებით **20...30 მწმ**-ის ხანგრძლივობის მქონე სეგმენტებისათვის ვიწროზოლოვანი ფილტრების ნაკრების საშუალებით განისაზღვრება ამპლიტუდური სპექტრი. რაც უფრო მეტია ფილტრების რაოდენობა, მით უკეთ ფასდება სპექტრი, მაგრამ მით უფრო მეტი ბიტია საჭირო მისი კოდირებისათვის და, შედეგად, მით მეტია კოდის საბოლოო სიჩქარე. ზოლური ფილტრების (**ზფ**) გამოსასვლელებზე მიღებული სიგნალები დეტექტირდება (გაიმართება), გატარდება დაბალი სიხშირეების ფილტრებში (**ღსფ**), დისკრეტიზირდება და ორობითად კოდირდება. უკანასკნელი პროცედურები ხორციელდება ანალოგურ-ციფრულ გარდამსახებში (**აცბ**) (ნახ. 2.26).

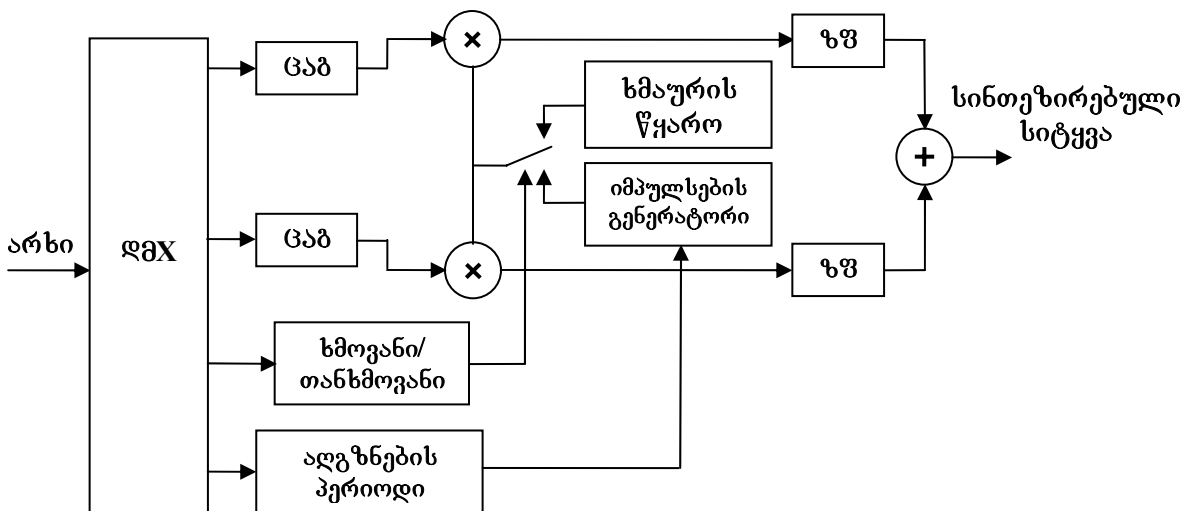


ნახ. 2.26. საარხო ვოკოდერის გადამცემი ნაწილი

ასეთნაირად განისაზღვრება ხმის წარმომქმნელი ტრაქტის მდორედ ცვლადი პარამეტრები და, გარდა ამისა, ძირითადი ტონისა და ხმოვანი ბგერების დეტექტორების საშუალებით – ალგზნების ძირითადი ტონის პერიოდი და ნიშანი იმის შესახებ, რომ ბგერა ხმოვანია თუ თანხმოვანი.

საარხო ვოკოდერი შეიძლება რეალიზებული იყოს როგორც ციფრული, ასევე ანალოგური ფორმის სახით და იგი უზრუნველყოფს სიტყვების საკმარის გარჩევადობას იმ დროს, როდესაც კოდის სიჩქარე 2,4 კბიტი/წმ-ის რიგისაა.

დეკოდერი (ნახ. 2.27) მიიღებს რა კოდერის მიერ გამომუშავებულ ინფორმაციას, ამ უკანასკნელს დაამუშავებს უკუმიმართულებით, რის შედეგადაც მის გამოსასვლელზე მიიღება მეტყველების სიგნალი, რომელიც გარკვეული დოზით მსგავსია საწყისი სიგნალისა.



ნახ. 2.27. საარხო ვოკოდერის მიმღები ნაწილი

მოდელის სიმარტივის გამო ძნელი წარმოსადგენია, რომ მივიღოთ აღდგენილი სიგნალის კარგი ხარისხი. მართლაც, საარხო ვოკოდერები ძირითადად გამოიყენება მხოლოდ იქ, სადაც უმთავრესია გარჩევადობა და შეკუმშვის მაღალი ხარისხი, კერძოდ: კავშირის სამხედრო სახეობაში, ავიაციაში, კოსმოსურ კავშირში და ა.შ.

ჰომომორფული ვოკოდერი

სიგნალების ჰომომორფული დამუშავება წარმოადგენს მათი არაწრფივი დამუშავების მეთოდების ერთერთ სახეობას, რომელიც შეიძლება ეფექტურად

იქნას გამოყენებული რთული სიგნალებისათვის, კერძოდ კი – მეტყველების სიგნალებისათვის.

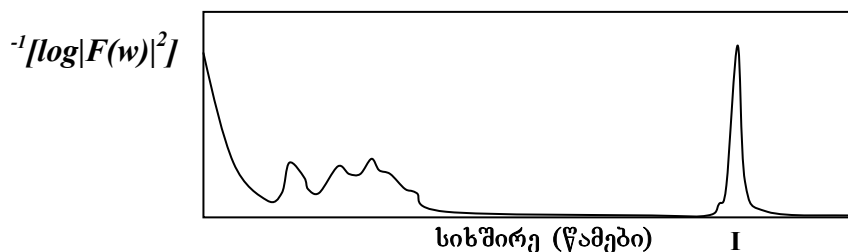
ვოკოდერებში ხმის წარმომქმნელი სისტემის მოდელის გამოყენების გათვალისწინებით მეტყველების სიგნალი შეიძლება წარმოვადგინოთ როგორც ხმის ტრაქტის იმპულსური გარდამავალი მახასიათებლის დროითი ხვევა ადგზნების სიგნალთან. სიხშირულ არეში ეს შეესაბამება ხმის ტრაქტის სიხშირული მახასიათებლისა და ადგზნების სიგნალის სპექტრის ნამრავს. ამ ნამრავლის გალოგარიტმების შედეგად მივიღებთ ადგზნების სიგნალისა და ხმის ტრაქტის სიხშირული მახასიათებლის სპექტრის ლოგარიტმების ჯამს. იმის გამო, რომ ადამიანის ყური პრაქტიკულად არამგრძობიარეა სიგნალის ფაზის მიმართ, შეიძლება გამოვიყენოთ მხოლოდ ამპლიტუდური სპექტრები:

$$\log(|S(e^{j\omega})|) = \log(|P(e^{j\omega})|) + \log(|V(e^{j\omega})|), \quad (2.24)$$

სადაც $S(e^{j\omega})$ და $P(e^{j\omega})$ – მეტყველებისა და ადგზნების სიგნალის სპექტრებია, ხოლო $V(e^{j\omega})$ – ხმის ტრაქტის სიხშირული მახასიათებელია.

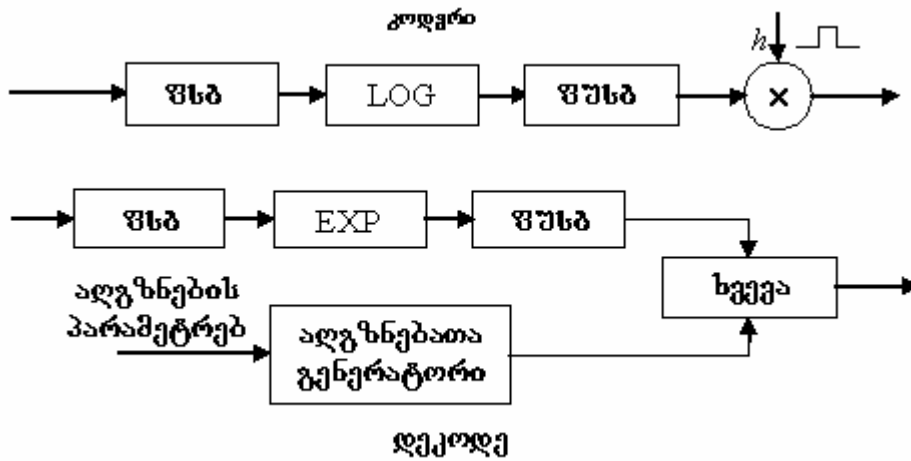
თუ ახლა $\log(|S(e^{j\omega})|)$ -ს გარდავქმნით ფურიეს უკუგარდასახვის გამოყენებით, მაშინ მივიღებთ სიგნალის ეგრეთ წოდებულ კეპსტრს.

ხმის სიგნალის პარამეტრები დროში შედარებით მდორედ იცვლება (მათი სპექტრი მოთავსებულია დაბალი სიხშირეების არეში) მაშინ, როდესაც ადგზნების სიგნალი სწრაფად ოსცილირებადი ფუნქციაა (მისი სპექტრი თავმოყრილია მაღალი სიხშირეების არეში). ამის გამო მეტყველების სიგნალის კეპსტრის ეს შემდგენები შეიძლება დავეოთ (ნახ. 2.28) და მათი კოდირება მოვახდინოთ ცალ-ცალკე.



ნახ. 2.28. მეტყველების სიგნალის სპექტრი

მეტყველების ჰომომორფული კოდერი/დეკოდერის სქემა წარმოდგენილია 2.29 ნახაზზე. ამ სქემის გამოყენებისას შეიძლება კოდის სიჩქარის შემცირება დაახლოებით 4 კბიტი/წმ-დე.



ნახ. 2.29. ჰომომორფული კოდერი/დეკოდერი

ფორმანტული ვოკოდერები

როგორც იყო აღნიშნული ადრე, მეტყველების სიგნალის შესახებ ძირითადი ინფორმაცია მოთავსებულია მისი ფორმანტების შემდგენების ზოლში. თუ მაღალი სიხუსტით მოვახდენთ ამ ფორმანტების პარამეტრების განსაზღვრასა და კოდირებას, მაშინ მივიღებთ კოდის ძალიან მცირე საბოლოო სიჩქარეს –

1 კბიტი/წმ-ზე ნაკლებს. სამწუხაროდ, ამის პრაქტიკული განხორციელება ძალზე ძნელია, რის გამოც მეტყველების ფორმანტული ვოკოდერები ჯერჯერობით ფართოდ არაა გავრცელებული.

ვოკოდერები წრფივი წინასწარმეტყველებით

წრფივი წინასწარმეტყველების საფუძველზე მომუშავე ვოკოდერები იყენებენ ხმის წარმომქმნელი სისტემის ისეთივე მიდელს, როგორც დანარჩენი უკვე განხილული ვოკოდერები. მათთვის განმასხვავებელია მხოლოდ ტრაქტის პარამეტრების განსაზღვრის მეთოდი. ასეთი კოდერებისათვის ხმის ტრაქტი მიჩნეულია უწყვეტი იმპულსური გარდამავალი მახასიათებლის მქონე წრფივ ფილტრად, რომელშიც სიგნალის ყოველი მომდევნო მნიშვნელობა შეიძლება მიღებული იყოს მისი წინამდებარე გარკვეული რაოდენობის მნიშვნელობების წრფივი კომბინაციით.

აღნიშნულ ვოკოდერებში მეტყველების სიგნალი დაიყოფა დაახლოებით 20 მწმ ხანგრძლივობის ბლოკებად, რომელთაგან თითოეულისათვის განისაზღვრება წინასწარმეტყველების ფილტრის კოეფიციენტები. ეს კოეფიციენტები იკვანტება და გადაიცემა დეკოდერისაკენ. შემდეგ მეტყველების სიგნალს ატარებენ ფილტრში, რომელსაც გააჩნია ხმის ტრაქტის მახასიათებლის შებრუნებული (უკუ) სიხშირული მახასიათებელი. ფილტრის გამოსასვლელზე მიიღება წინასწარმეტყველების შეცდომა. წინასწარმეტყველების მოწყობილობის დანიშნულებაა კორელაციის მოცილება სიგნალის მეზობელ ანათვლებს შორის. შედეგად სიგნალში გაცილებით მკვეთრად ვლინდება ხანგრძლივი კორელაცია, რაც იძლევა ძირითადი ტონის სიხშირის უფრო ზუსტად განსაზღვრისა და ხმოვანი/თანხმოვანი ბგერების მიმანიშნებლის გამოყოფის საშუალებას.

დღეისათვის წრფივი წინასწარმეტყველების საფუძველზე მომუშავე ვოკოდერები ძალზე პოპულარულია, ვინაიდან მათ მიერ გამოყენებული ხმის ტრაქტის ყველა ფილტრული მოდელი მუშაობს ძალიან კარგად. ასეთი კოდერების საშუალებით მიღებული კოდების სიჩქარე მეტყველების დამაკმაყოფილებელი ხარისხის შემთხვევაში აღწევს **2,4 კბიტი/წმ-ს**.

2.10.3. მეტყველების სიგნალების კოდირების ჰიბრიდული მეთოდები

მეტყველების კოდირების ჰიბრიდული ანუ კომბინირებული მეთოდები ავსებს შუალედს *სიგნალის ფორმის კოდერებს* (რომლებიც სრულებით არ ითვალისწინებენ სიგნალის ბუნებას) და *წყაროს კოდერებს* (რომლებიც არსებითად ახორციელებენ არა სიგნალის, არამედ ამ სიგნალის წარმომქმნელი წყაროს მოდელის პარამეტრების კოდირებას) შორის. როგორც ადრე იყო აღნიშნული, სიგნალის ფორმის კოდერები უზრუნველყოფენ მეტყველების ძალიან კარგ ხარისხს კოდირების 16 კბიტი/წმ-ზე უფრო მაღალი სიჩქარეების შემთხვევაში, მაგრამ საერთოდ არ მუშაობენ უფრო დაბალ სიხშირეებზე მაშინ, როდესაც ვოკოდერები უზრუნველყოფენ მეტყველების გარჩევადობას **2,4 კბიტი/წმ** და უფრო მცირე სიჩქარეების შემთხვევაშიც კი, მაგრამ ისინი ვერ იძლევიან კარგ ხარისხს კოდის ნებისმიერი სიჩქარისათვის.

დღეისათვის უფრო გავრცელებულია კოდირების ჰიბრიდული მეთოდები, რომლებიც მუშაობენ დროით სივრცეში (ანუ მუშაობენ უშუალოდ სიგნალზე და არა მის სპექტრზე ან სხვა წრფივ გარდასახვებზე). ეს მეთოდები ეფუძნება სიგნალის ანალიზს მისი სინთეზის ბაზაზე (ეგრეთ წოდებული **ABS**–კოდეკები).

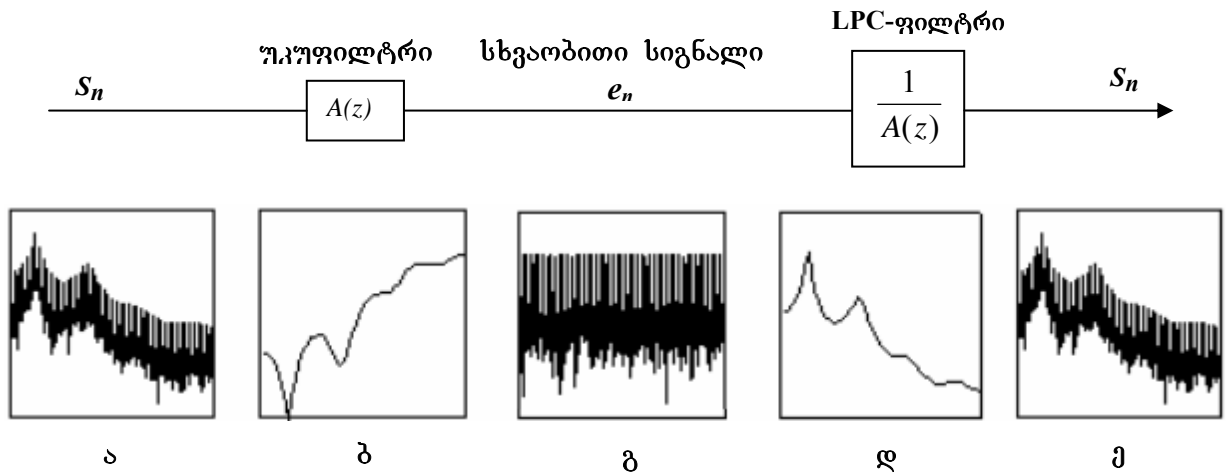
ეს კოდერები, ისევე როგორც ვოკოდერები, იყენებენ ხმის ტრაქტის მოდელს, ოღონდ რამდენადმე განსხვავებულად – ალგზნების სიგნალის შერჩევა ხდება ისეთნაირად, რომ უზრუნველყოფილია მეტყველების სინთეზირებული და საწყისი სიგნალების საუკეთესო თანხვედრა.

ABS-კოდერები პირველად წარმოდგენილი იყო არც თუ დიდი ხნის წინათ (1982 წელს) და მათ პირველ სახეობას უწოდეს **MPE**-კოდერები (**Multi-Pulse Excited**- კოდერები მრავალიმპულსიანი ალგზნებით). შემდგომში შემოთავაზებული იყო უფრო სრულყოფილი **RPE**-კოდერები (**Regular-Pulse Excited** – კოდერები რეგულარული იმპულსური ალგზნებით) და **CELP**-კოდერები (**Codebook-Excited Linear Predictive** – კოდერები ალგზნებით კოდური წიგნაბის გამოყენების საფუძველზე). დღეისათვის არსებობს მათი სხვა სახეობებიც, მაგრამ ყველა მათგანი იყენებს საერთო იდეას.

იმისათვის, რომ გავიგოთ, თუ რა ემყარება **ABS**-კოდერის ეფექტურობა და ხარისხი, პირველ რიგში განვიხილოთ ეგრეთ წოდებული **RELP**-კოდერის (**Residual Excited Linear Prediction - RELP**) მუშაობა.

თუ მეტყველების სიგნალს, რომელსაც გააჩნია 2.30ა ნახაზზე წარმოდგენილი სპექტრი, გავატარებთ წრფივ წინასწარმეტყველში, რომელსაც აქვს 2.30ბ ნახაზზე ნაჩვენები სისშირული მახასიათებელი, მაშინ გამოძვალა სიგნალის ანათვლებს შორის არსებული (*წინასწარმეტყველების შეცდომის*) კორელაცია მნიშვნელოვნად შემცირდება. თუ წინასწარმეტყველება შესრულდება საკმაოდ კარგად, მაშინ წინასწარმეტყველის გამოსასვლელზე პრაქტიკულად მივიღებთ თეთრ ხმაურს, რომელსაც გააჩნია თანაბარი სპექტრი (ნახ. 2.26გ).

ამასთანავე თეთრი ხმაური (*წინასწარმეტყველების შეცდომა*) შეიცავს სრულ ინფორმაციას საკოდირებელი სიგნალის შესახებ და თუ მას ხელახლა გავატარებთ **LPC**-ფილტრში, რომლის სისშირული მახასიათებელი წარმოდგენილია 2.30დ ნახაზზე, მაშინ ჩვენ აბსოლუტურად ზუსტად აღვადგენთ მეტყველების საწყის სიგნალს. ვინაიდან ეს ინფორმაცია წინასწარმეტყველების შეცდომის სპექტრში მეტნაკლებად განაწილებულია თანაბრად, ამიტომ წარმოიშვა წინასწარმეტყველების **E(ω)** შეცდომის სპექტრის მხოლოდ მცირე ნაწილის კოდირებისა და გადაცემის, ხოლო სპექტრის დანარჩენი ნაწილის დეკოდერში აღდგენის იდეა.



ნახ. 2.30. მეტყველების **RELP**-კოდერის სქემა და შესაბამისი სპექტრები და მახასიათებლები

RELP-კოდერში წინასწარმეტყველების შეცდომის სიგნალი გატარდება დაბალსიხშირულ ფილტრში, რომლის ჩამოჭრის სიხშირეა დაახლოებით 1 კჰც. ფილტრის გამოსასვლელზე მიღებული სიგნალი კოდირდება ფორმის მიხედვით, მაგალითად, **დოკმ**-კოდერის გამოყენებით. დეკოდერში წინასწარმეტყველების შეცდომა აღდგება სიხშირეთა დაშორებულ არეში, რომელშიც ის იყო გადატანილი კოდერის მიერ.

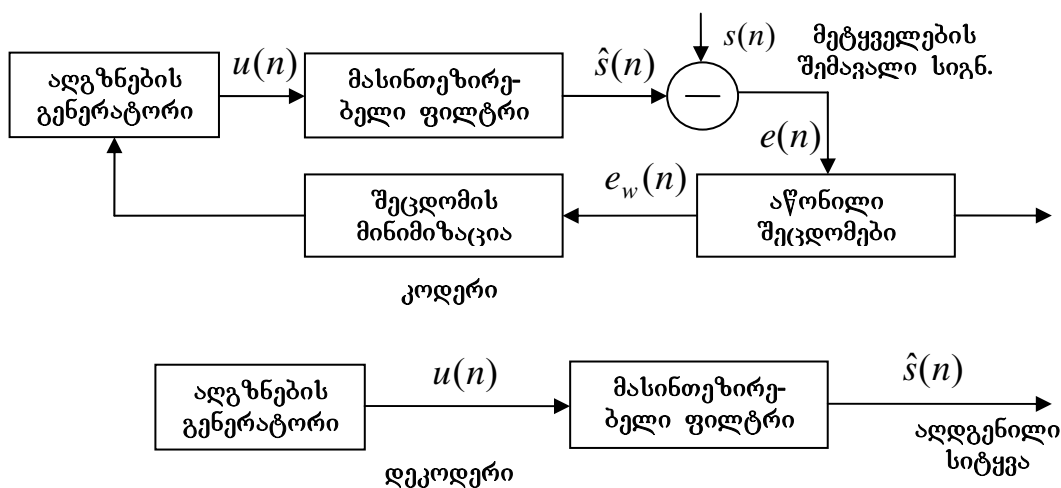
RELP-კოდერი იმუშავებდა იდეალურად, წრფივი წინასწარმეტყველების პროცესში რომ ვლტებულობდეთ თეთრ ხმაურს. თუმცა მეტყველების სიგნალში კვაზიპერიოდული ფორმანტული შემდგენების არსებობის გამო წრფივ წინასწარმეტყველს არ შეუძლია ფორმანტების ძირითადი ტონის პერიოდიანი ხანგრძლივი კორელაციის მოცილება და ისინი ნამდვილად იარსებებენ წინასწარმეტყველების შეცდომის სპექტრში. თუ ამჯერად $E(\omega)$ -ს გავატარებთ დაბალი სიხშირეების ფილტრში, მაშინ დაიკარგება მაღალსიხშირული ფორმანტული შემდგენები და მათი შემდგომში აღდგენა იქნება შეუძლებელი.

RELP-კოდერები იძლევიან სიგნალის არც თუ ისე ცუდ ხარისხს კოდის 9,6 კბიტი/წმ სიჩქარის შემთხვევაში, თუმცა მათთვის გარკვეული დოზით დამახასიათებელია ვოკოდერების ნაკლოვანება – აღდგენილი მეტყველების სინთეტიკური ხასიათი. ამის გამო აღნიშნული კოდერები ყველგან შეცვალა ანალოგიური პრინციპით მომუშავე სხვადასხვა სახეობის **ABS**-კოდერებმა.

ABS-კოდერი მუშაობს შემდგენაირად. საკოდირებელი შემავალი სიგნალი ციფრულ ფორმაში წარმოდგენილი მისი ანათვლების სახით დაიყოფა

დაახლოებით 20 მწმ ხანგრძლივობის ფრაგმენტებად, რომელთა საზღვრებში სიგნალის თვისებები იცვლება უმნიშვნელოდ. თითოეული ფრაგმენტისათვის განისაზღვრება მასინთეზირებელი ფილტრის (ხმის ტრაქტის ანალოგის) მიმდინარე პარამეტრები და შემდგომ აირჩევა ადგზნების სიგნალი, რომელიც, გატარებულია რა მასინთეზირებელ ფილტრში, მინიმუმამდე ამცირებს განსხვავებას (შეცდომას) შემაჯავალ და სინთეზირებულ სიგნალებს შორის.

ამრიგად, კოდირების მეთოდის დასახელება *Analysis-by-Synthesis* იმაში მდგომარეობს, რომ კოდერი ანალიზებს შემაჯავალ სიტყვას მრავალი მასთან მიახლოებული სიგნალების სინთეზის საშუალებით. საბოლოოდ კოდერი დეკოდერისაკენ გასცემს ინფორმაციას, რომელიც წარმოადგენს მასინთეზირებელი ფილტრისა და ადგზნების სიგნალის მიმდინარე პარამეტრების კომბინაციას. სასურველია, რომ ამ მონაცემების რაოდენობა იყოს რაც შეიძლება მცირე. ა. პარამეტრების მიხედვით დეკოდერი აღადგენს კოდირებულ სიტყვას, თანაც ამას აკეთებს ისეთნაირადვე, როგორც აკეთებდა კოდერი ანალიზის დროს. სხვადასხვა ტიპის **ABS**-კოდერებს შორის განსხვავება მდგომარეობს თითოეულ მათგანში მასინთეზირებელი ფილტრის ადგზნების $u(n)$ სიგნალის შერჩევაში. თეორიულად მასინთეზირებელი ფილტრის შესასვლელს უნდა მიეწოდოს ადგზნების სხვადასხვა სიგნალების დიდი რაოდენობა იმისათვის, რომ ინახოს, თუ რომელი სიგნალი მიიღება მის გამოსასვლელზე და იგი შედარდეს საკოდირებელ სიგნალს. ადგზნების სიგნალი, რომელიც მოგვცემს მინიმალურ აწონილ შეცდომას ორინალსა და სინთეზირებულ სიტყვებს შორის, შეირჩევა როგორც კოდირების შედეგი. ადგზნების სიგნალის განსაზღვრის სწორედ ეს ჩაკეტილი სქემა (ნახ. 2.31) უზრუნველყოფს მეტყველების მაღალ ხარისხს კოდის მცირე სიჩქარეების პირობებში.



ნახ. 2.31. **ABS** კოდერისა და დეკოდერის სქემები

პრობლემა მდგომარეობს შესასრულებელი ოპერაციების დიდ რაოდენობაში, რომლებიც საჭიროა ალგზნების საუკეთესო სიგნალის შერჩევისათვის. მაგრამ გამოთვლითი და მიკროპროცესორული ტექნიკის დღევანდელი შესაძლებლობების პირობებში ეს სრულიად მიღწევადია.

მრავალმპულსური კოდერები (MPE-კოდერები)

როგორც უკვე იყო აღნიშნული, წინასწარმეტყველების ფილტრში მეტყველების სიგნალის გავლისას მნიშვნელოვნად მცირდება კორელაცია მეზობელ ანათვლებს შორის. თუმცა ხმოვანი ბგერებისათვის ფორმანტული შემდგენების არსებობა იწვევს მეტყველების სიგნალში კვაზიპერიოდულობისა და ხანგრძლივი მაღალი კორელაციის წარმოქმნას. ეს პერიოდულობა არ აღმოიფხვრება წრფივი წინასწარმეტყველებით და იგი წინასწარმეტყველების სიგნალში იწვევს მაღალამპლიტუდური ნარჩილების წარმოქმნას.

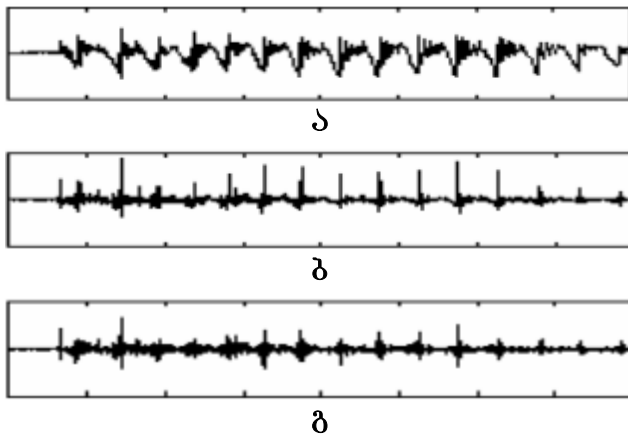
ხანგრძლივდროიანი კორელაციის მოცილებისათვის წინასწარმეტყველების შეცდომის სიგნალი შეიზლება გავატაროთ მეორე წრფივ წინასწარმეტყველში, რომელმაც უნდა აღმოიფხვრას კორელაცია მეტყველების სიგნალის არა მეზობელ ანათვლებს შორის, არამედ წინასწარმეტყველების შეცდომის მეზობელ პერიოდებს შორის. ეს მიიღწევა წინასწარმეტყველში მეტყველების სიგნალის ძირითადი ტონის პერიოდით დროითი დაყოვნების განხორციელების გზით.

$$P(z) = 1 - \sum_i \beta_i z^{-M-i}, \tag{2.25}$$

სადაც M – ძირითადი ტონის პერიოდია.

2.32 ნახაზზე ნაჩვენებია: ა – მეტყველების საწყისი სიგნალი; ბ – მოკლე-დროიანი წინასწარმეტყველების შეცდომის სიგნალი (3-ჯერ გადიდებული); გ – სიგნალი ორკასკადა (მცირედროიანი+ხანგრძლივდროიანი) წინასწარმეტყველის

გამოსასვლელზე (10-ჯერ გადიდებული).



ნახ. 2.32. საწყისი და წინასწარმეტყველი მოწყობილობების გამომავალი სიგნალები

თუ ახლა წინასწარმეტყველების შეცდომის რეზურტირებულ სიგნალს მივაწვდით ერთმანეთთან მიმდევრობით შეერთებულ მცირე და დიდი ხანგრძლივობის ფილტრებს—წინასწარმეტყველების მოწყობილობებს, მაშინ გამოსასვლელზე მივიღებთ მეტყველების საწყის დაუმახინჯებელ სიგნალს. შეიძლება და წინასწარმეტყველების შეცდომის მიღებული სიგნალის კოდირება და კავშირის არხში გადაცემა და ეს უზრუნველყოფდა კიდევ გარკვეულ ეკონომიას მისი მნიშვნელოვნად მცირე ამპლიტუდის გამო მეტყველების საწყის სიგნალთან შედარებით. თუმცა სიგნალის ფორმა (ნახ. 2.32გ) მაინც საკმაოდ რთული რჩება, რაც მისი კოდირებისათვის მოითხოვს ბიტების საკმაოდ დიდ რაოდენობას.

მრავალიმპულსურ MPE კოდერებში ადგზნების $u(n)$ სიგნალის როლში იღებენ არა წინასწარმეტყველების შეცდომას (ნახ. 2.32გ), არამედ *ოთხი-ექვსი იმპულსისაგან შედგენილ მიმდევრობას*. თითოეული ამ იმპულსის დროითი მდებარეობა და მათი ამპლიტუდები განისაზღვრება *სინთეზის გავლით ანალიზის (ABS)* პროცედურის პროცესში მანამ, სანამ მეტყველების საწყის და სინთეზი-რებულ სიგნალებს შორის არ მიიღწევა მინიმალური განსხვავება. ადგზნების იმპულსების პარამეტრებს, რომლებითაც ხდება შეცდომის მინიმიზაცია, არჩევენ მიმდევრობით – თავიდან პირველი იმპულსისათვის, შემდეგ მეორისათვის და ა.შ.

პრაქტიკაში საკმარისია იმპულსების მდებარეობის აღება დაახლოებით

1 მწმ-ის ბიჯითა და 5%-იანი სიზუსტით. ამ შემთხვევაში უზრუნველყოფილია მეტყველების კარგი ხარისხი კოდის 10 კბიტ/წმ სიჩქარისას. მეტყველების სიგნალის 20 მწმ-იანი ფრაგმენტისათვის გამოიყენება ადგზნების 6 იმპულსი. თითოეული მათგანის მდებარეობა მოცემულია 1 მწმ-ის სიზუსტით, რომელიც ფრაგმენტის ხანგრძლივობის 1/20-ის ტოლია და რომელიც, თავის მხრივ, შეესაბამება 5 ბიტს იმპულსზე. იმპულსის ამპლიტუდა მოცემულია 5%-იანი სიზუსტით, რომელიც უდრის 5 ბიტს იმპულსზე. ამის შედეგად ვღებულობთ ადგზნების სიგნალის კოდის მინიმალურ სიჩქარეს – $6 \times 10 = 60$ ბიტ/20მწმ. გარდა ამისა, კოდს უნდა დაემატოს მოკლე და დიდი ხანგრძლივობის წინასწარმეტყველების ფილტრების მოცემული ფრაგმენტისათვის გაანგარიშებული პარამეტრები, რაც დაახლოებით შეადგენს 80...100 ბიტს 20მწმ-ზე. შედეგად მიიღება კოდის სიჩქარე 160 ბიტ/20 მწმ=8 კბიტ/წმ.

კოდერები რეგულარული იმპულსური აღვზნებით (RPE-კოდერები)

ისევე როგორც **MPE-კოდეკი**, **Regular Pulse Excited**, ანუ **RPE-კოდეკი**, აღვზნების $u(n)$ სიგნალად იყენებს მოკლე იმპულსების ფიქსირებულ ნაკრებს. თუმცა ამ კოდეკს იმპულსები განლაგებულია რეგულარულად ერთმანეთისაგან თანაბარ მანძილზე და კოდერს მოეთხოვება მხოლოდ პირველი იმპულსის მდებარეობისა და ყველა იმპულსის ამპლიტუდის განსაზღვრა. ამრიგად, დეკოდერს უნდა გადაეცეს უფრო მცირე ინფორმაცია იმპულსების მდებარეობის შესახებ, რის გამოც აღვზნების სიგნალში შეიძლება ჩაერთოს მათი მეტი რაოდენობა და ამით გაუმჯობესდეს სინთეზირებული სიგნალის მიახლოება ორიგინალთან. მაგალითად, კოდის 10 კბიტი/წმ სიჩქარის შემთხვევაში თუ გამოიყენება აღვზნების ოთხიმპულსიანი სიგნალი, მაშინ **RPE-კოდეკში** უკვე შეიძლება გამოვიყენოთ ათიმპულსიანი სიგნალი. ამასთანავე მნიშვნელოვნად უმჯობესდება მეტყველების ხარისხი.

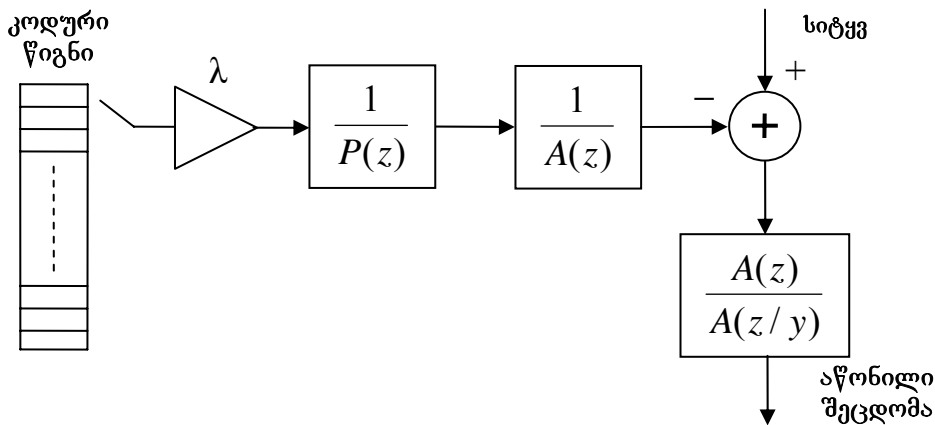
რეგულარული იმპულსური აღვზნების RPE მეთოდი დღეს ფართოდ გამოიყენება კავშირის სისტემებში, მათ შორის ფიჭური კავშირის GSM სისტემაში.

კოდერები კოდური წიგნების საფუძველზე აღვზნებით (CELP-კოდერები)

კოდირების **MPE** და **RPE** მეთოდები უზრუნველყოფენ მეტყველების კარგ ხარისხს, თუ კოდის სიჩქარეა 10 კბიტი/წმ და მეტი, მაგრამ ისინი ძლიერ ამახინჯებენ სიგნალს უფრო დაბალი სიჩქარეების შემთხვევაში. საქმე იმაშია, რომ აღვზნების სიგნალის აუცილებელი პარამეტრების (იმპულსების დროითი მდებარეობა და ამპლიტუდა) საჭირო სიზუსტით აღწერისათვის უბრალოდ საკმარისი არაა ბიტების რაოდენობა.

ზემოაღნიშნულის გამო დამუშავდა მეთოდი, რომელიც აღვზნების სიგნალად იყენებს არა იმპულსების მიმდევრობას, რომელიც მოცემულია თავისი პარამეტრების ნაკრებით, არამედ სხვადასხვა ფორმის აღვზნების სიგნალების ბიბლიოთეკას (კოდურ წიგნებს), რომლებიც სპეციალურადაა მომზადებული და ჩაწერილი მახსოვრობის მოწყობილობაში – **Codebook Excited Linear Prediction (CELP)**.

აღზნების სიგნალების ფორმირების **CELP**-კოდერის სქემა ნაჩვენებია 2.33 ნახაზზე.



ნახ. 2.33. **CELP**-კოდერის სქემა

ამ შემთხვევაში კოდირების შედეგია არა აღზნების სიგნალების იმპულსების პარამეტრები, არამედ კოდური წიგნის ინდექსი (აღზნების სიგნალის მასში შენახული ნიმუშის ნომერი) და, აგრეთვე, მისი ამპლიტუდა. თუ კოდური წიგნი შეიცავს, მაგალითად, 1024 სიგნალს, ხოლო სიგნალის ამპლიტუდის კოდირება ხდება 2...3%-იანი სიზუსტით, მაშინ 20 მწმ-იანი ფრაგმენტისათვის ბიტების აუცილებელი რაოდენობაა 10 (ინდექსისათვის) + 5 (ამპლიტუდისათვის)=15 ბიტი. შედარებისათვის ავლნიშნოთ, რომ **GSM RPE**-კოდერის შემთხვევაში გამოიყენება 47 ბიტი. მართალია, რომ წარმოდგენილი ალგორითმის საფუძველზე კოდირების პროცედურა მოითხოვს დიდი რაოდენობის გაანგარიშებებს, რის გამოც **CELP**-კოდერების რეალიზაცია შესაძლებელი აღმოჩნდა მხოლოდ უკანასკნელ პერიოდში სპეციალიზირებული სასიგნალო პროცესორების გამოყენებით, რომელთა მწარმოებლურობაა წამში 300 მილიონი და მეტი რაოდენობის ოპერაცია.

CELP ალგორითმის საფუძველზე განხორციელებული კოდირება წარმატებით გამოიყენება კავშირის თანამედროვე სისტემებში, რომელთა კოდების შესაბამისი სიჩქარეებია 16-დან 4,8 კბიტი/წმ-დე. ამასთან 16 კბიტი/წმ-ის ტოლი სიჩქარისათვის **CELP** უზრუნველყოფს მეტყველების ისეთივე ხარისხს, რომელიც მიიღება 64 კბიტი/წმ სიჩქარიანი **PCM**-ის დროს, ხოლო 4,8 კბიტი/წმ სიჩქარის შემთხვევაში მეტყველების ხარისხი ისეთივეა, როგორც 13 კბიტი/წმ სიჩქარიანი **GSM RPE** სისტემის გამოყენების დროს.

შეტყობინებათა წყაროს ენტროპია, დაკვიანება და დაკვიანების შეცდომები. შესაბამისი ალგორითმების რეალიზაციის პროგრამა

a) For Image processing

Insert Image name -- name = "LENA.bmp" <-----

LIST of IMAGE's
 Ships.bmp
 School.bmp
 Rose.bmp
 LENA.bmp

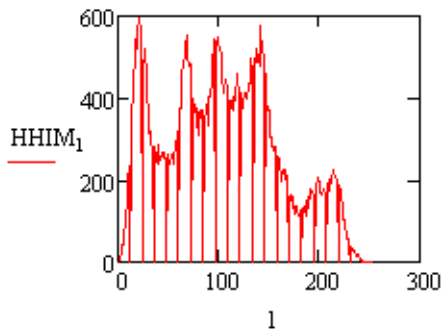
```
IM := READBMP(name)
```



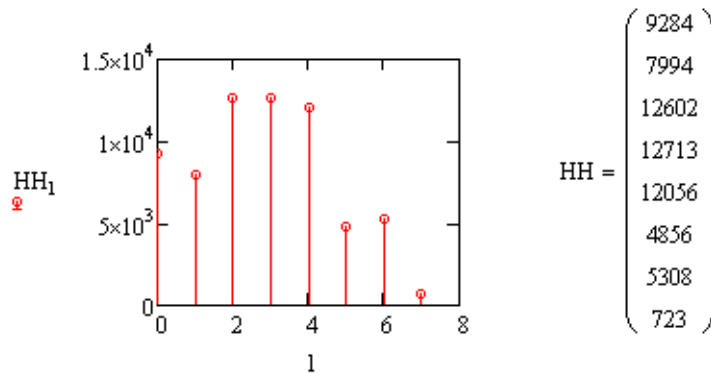
IM =

	0	1	2	3	4
0	137	136	133	136	138
1	137	136	133	136	138
2	138	133	134	134	136
3	133	133	133	130	134
4	129	133	130	130	133
5	131	133	130	122	132
6	131	130	130	130	132
7	131	132	130	130	131
8	129	130	130	127	130
9	129	130	132	129	131
10	130	130	129	128	131
11	131	129	125	130	132
12	129	127	127	130	...

```
IM
mean(IM) = 99
l := 0..255
HHIM := hist(256,IM)
```



```
ur := 8 number level for Quantization
l := 0..ur - 1
HH := hist(ur,IM)
```



```

ENT(pict,lev) :=
  sh ← 0
  for i ∈ 0..lev - 1
    pi ←  $\frac{\text{pict}_{i,0}}{256 \cdot 256}$ 
    ni ← -pi · log(pi,2) if pi ≠ 0
    ni ← 0 otherwise
    sh ← ni + sh
  sh

```

ENT(HHIM,256) = 7.568

ENT(HH,ut) = 2.779

b) For Signal processing

$N := 100$

mini := -10

maxi := 70

maxi - mini = 80 max level quantization

ceil(log(maxi - mini,2)) = 7 Entropia - max

k := 0..N - 1

$v := \text{round}(\text{runif}(N, \text{mini}, \text{maxi}))$

qwan := 8 new level qwantization

q := 0..qwan - 1

ceil(log(qwan,2)) = 3 Entropia - max after qwant

stepi := round($\frac{\text{maxi} - \text{mini}}{\text{qwan}}$)

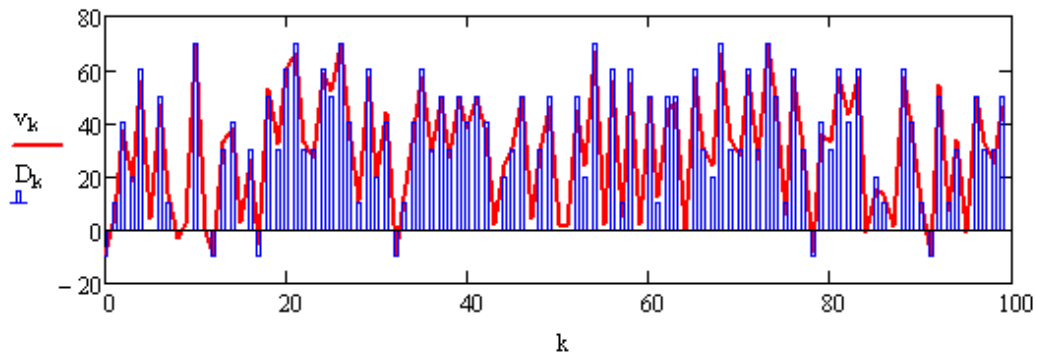
stepi = 10

$D_k := \text{round}\left(\frac{v_k}{\text{stepi}}\right) \cdot \text{stepi}$

$$v^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & -10 & 5 & 37 & 18 & 56 & 4 & 47 & 14 & -3 & \dots \\ \hline \end{array}$$

$$D^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & -10 & 10 & 40 & 20 & 60 & 0 & 50 & 10 & 0 & \dots \\ \hline \end{array}$$

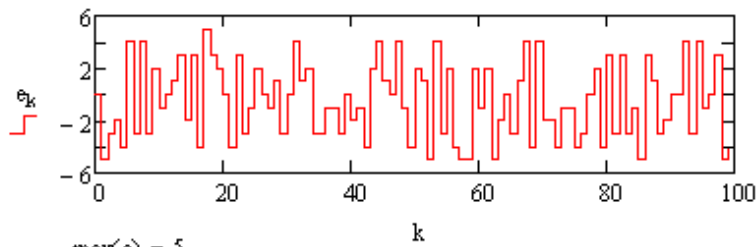
length(D) = 100



v_k - original signal

D_k - quantiz signal

$e_k := v_k - D_k$ error quantization

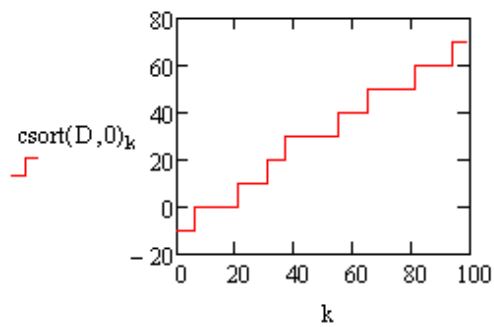


$\max(e) = 5$

$\min(e) = -5$

$e_k =$

0
-5
-3
-2
-4
4
-3
4
-3
2
-1
0
1
3
-2
...



$\text{csort}(D,0) =$

	0
0	-10
1	-10
2	-10
3	-10
4	-10
5	-10
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	...

დისკრეტული კოსინუსური გარდასახვის რეალიზაციის პროგრამა

$$DCP := \begin{pmatrix} 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 \\ 0.490 & 0.416 & 0.278 & 0.098 & -0.098 & -0.278 & -0.416 & -0.490 \\ 0.462 & 0.191 & -0.191 & -0.462 & -0.462 & -0.191 & 0.191 & 0.462 \\ 0.416 & -0.098 & -0.490 & -0.278 & 0.278 & 0.490 & 0.098 & -0.416 \\ 0.354 & -0.354 & -0.354 & 0.354 & 0.354 & -0.354 & -0.354 & 0.354 \\ 0.278 & -0.490 & 0.098 & 0.416 & -0.416 & -0.098 & 0.490 & -0.278 \\ 0.191 & -0.462 & 0.462 & -0.191 & -0.191 & 0.462 & -0.462 & 0.191 \\ 0.098 & -0.278 & 0.416 & -0.490 & 0.490 & -0.416 & 0.278 & -0.098 \end{pmatrix}$$

DCP_ 8x8 ზომის გარდასახვისათვის ბაზისური (კოსინუსური) კოეფიციენტების მატრიცაა

$$QW0 := \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad \text{დაკვანტვის ცხრილი}$$

Creating 1-th matrix QW1(8x8)

f := 0..7

FQ_f := 1

QW1 := augment(FQ,FQ,FQ,FQ,FQ,FQ,FQ,FQ)

$$QW1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

For DCP matrix 8x8

```
jm := cols(DCP)      jm = 8  cols in submatrix
im := rows(DCP)     im = 8  rows in submatrix
```

If want is multiplied the DCP matrix on any whole factor,
insert : 1, 2, 3, 4, ... nDP := 1

LIST of IMAGE's

WINTER1.bmp
Ships.bmp
School.bmp
Rose.bmp
LENA.bmp
LEAVES.bmp
BERRIES.bmp

Insert Image name -- name := "Lena.bmp"

PIC := READBMP(name)

```
cols(PIC) = 256      max(PIC) = 238
rows(PIC) = 256     min(PIC) = 3
```

Insert number of coefficient 1/NC for coding -- NC := 1 **NC= 256, 128, 64, 32, 16, 8, 4, 2, 1**

$$CNN := \frac{NC}{(im \cdot jm)} \quad \text{for decoding}$$

CNN = 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64

CNN = 0.016

Enter number for a choice such as a matrix of Qwantization 1 or 2

for DCP with Qwant -- 1 (QW)
non Qwantization -- 2 (QW1)

```
nDP = 1
p := 1
```

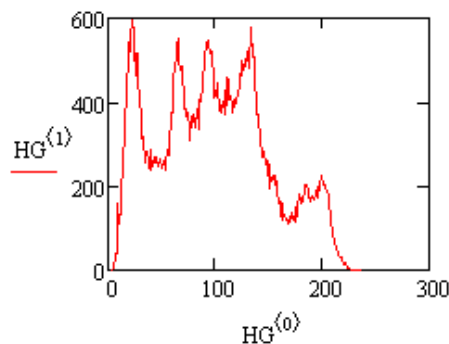
QWmatrix := 1

```
NMAT := nDP * DCP
qqw := QWmatrix
qqw = 1
QWQW := | JQW ← QWT if qqw = 1
         | JQW ← QW1 otherwise
         | JQW
```

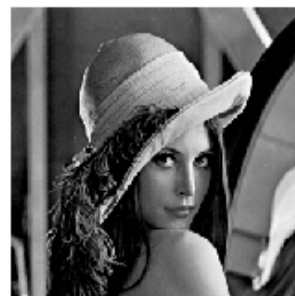
$$QWT := \begin{cases} \text{for } i \in 0..im - 1 \\ \text{for } j \in 0..jm - 1 \\ Q_{i,j} \leftarrow \text{trunc} \left(\frac{QW0_{i,j}}{p} \right) \\ Q \end{cases}$$

For Original IMAGE

HG := histogram(round(max(PIC) - min(PIC)), PIC)



original image
-- PIC



PIC

Coding PIC

For Matrix 8x8 -- submatrix 8x8

jm = 8 cols in submatrix
im = 8 rows in submatrix

```

MPAR :=
  for k ∈ 0..round( (rows(PIC) / im) ) - 1
    for l ∈ 0..round( (cols(PIC) / jm) ) - 1
      MZ ← submatrix[PIC, im·k, im·k + (im - 1), jm·l, jm·l + (jm - 1)]
      A ← (NMAT·MZ·NMATT) / NC
      for i ∈ 0..im - 1
        for j ∈ 0..jm - 1
          WM16im·k+i, jm·l+j ← trunc( (Ai,j) / (QWQWi,j) )
    end for
  end for
WM16
  
```

rows(MPAR) = 256

max(MPAR) = 104

cols(MPAR) = 256

min(MPAR) = -43

STATISTIC

```

ZNAKI :=
  pol ← 0
  otr ← 0
  for i ∈ 0..rows(MPAR) - 1
    for j ∈ 0..cols(MPAR) - 1
      pol ← pol + 1 if MPARi,j ≥ 0
      otr ← otr + 1 otherwise
  end for
pol
  
```

znPLUS = ZNAKI

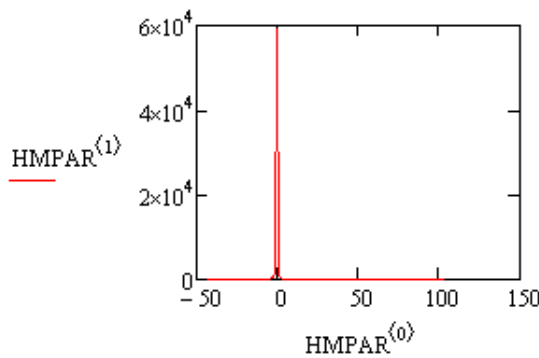
znPLUS = 62714

znMINUS := rows(MPAR)·cols(MPAR) - ZNAKI

znMINUS = 2822

$$\text{znPLUS} + \text{znMINUS} = 6.554 \times 10^4$$

HMPAR := histogram(trunc(max(MPAR) - min(MPAR)), MPAR)



$$\nabla \text{HMPAR}^{(1)} = 6.554 \times 10^4$$

	0	1
0	-42	2
1	-41	0
2	-40	1
3	-39	0
4	-38	0
5	-37	2
6	-36	2
7	-35	1
8	-34	2
9	-33	5
10	-32	1
11	-31	0
12	-30	2
13	-29	3
14	-28	0
15	-27	...

HMPAR =

$$\sum \text{HMPAR}^{(1)} = 65536$$

Entropia for MPAR

```

ENTR := | s ← ∑ HMPAR(1)
        | SH ← 0
        | for i ∈ 0..rows(HMPAR) - 1
        |   | PLi ←  $\frac{\text{HMPAR}_{i,1}}{s}$ 
        |   | HEi ←  $-PL_i \cdot \log_2(PL_i)$  if PLi ≠ 0
        |   | HEi ← 0 otherwise
        |   | SH ← HEi + SH
        | SH
    
```

ENTR = 0.969

MPAR =

	0	1	2	3	4	5	6	7	8	9
0	66	0	0	0	0	0	0	0	64	0
1	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	65	0	0	0	0	0	0	0	66	-1
9	0	0	0	0	0	0	0	0	-2	1
10	0	0	0	0	0	0	0	0	0	...

DECoding PIC

Global matrix MPAR 256x256

```

IMPIC := | for k ∈ 0..round( $\frac{\text{rows}(\text{MPAR})}{\text{im}}$ ) - 1
        |   for l ∈ 0..round( $\frac{\text{cols}(\text{MPAR})}{\text{jm}}$ ) - 1
        |     MZ ← submatrix[MPAR, im·k, im·k + (im - 1), jm·l, jm·l + (jm - 1)]
        |     for i ∈ 0..im - 1
        |       for j ∈ 0..jm - 1
        |         Ai,j ← MZi,j · QWQWi,j
        |         B ←  $\frac{64 \cdot [\text{NMAT}^{-1} \cdot A \cdot (\text{NMAT}^T)^{-1}]}{1}$  · CNN
        |     for i ∈ 0..im - 1
        |       for j ∈ 0..jm - 1
        |         | WMim·k+i, jm·l+j ← min(PIC) if Bi,j ≤ min(PIC)
        |         | WMim·k+i, jm·l+j ← max(PIC) if Bi,j ≥ max(PIC)
        |         | WMim·k+i, jm·l+j ← trunc(Bi,j) otherwise
        | WM
    
```

max(IMPIC) = 233

min(IMPIC) = 3

DECODER IMAGE

IMPIC =

	0	1	2	3	4	5
0	133	133	133	133	133	133
1	133	133	133	133	133	133
2	132	132	132	132	132	132
3	132	132	132	132	132	132
4	131	131	131	131	131	131
5	130	130	130	130	130	130
6	129	129	129	129	129	129
7	129	129	129	129	129	129
8	129	129	129	129	129	129
9	129	129	129	129	129	...



IMPIC

max(PIC) = 238

min(PIC) = 3

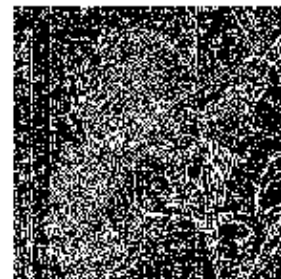
max(IMPIC) = 233

min(IMPIC) = 3

ERROR := PIC - IMPIC

ERROR =

	0	1	2	3	4	5
0	4	3	0	3	5	1
1	4	3	0	3	5	1
2	6	1	2	2	4	0
3	1	1	1	-2	2	1
4	-2	2	-1	-1	2	0
5	1	3	0	-8	2	1
6	2	1	1	1	3	2
7	2	3	1	1	2	2
8	0	1	1	-2	1	0
9	0	1	3	0	2	0
10	1	1	0	-1	2	...



ERROR

max(ERROR) = 74

min(ERROR) = -60

$$SNR := 10 \cdot \log \left[\frac{\sum_{m=0}^{255} \sum_{n=0}^{255} (PIC_{m,n})^2}{\sum_{m=0}^{255} \sum_{n=0}^{255} (PIC_{m,n} - IMPIC_{m,n})^2} \right]$$

$$PSNR := -10 \cdot \log \left[\frac{\sum_{m=0}^{255} \sum_{n=0}^{255} (PIC_{m,n} - IMPIC_{m,n})^2}{256^2 \cdot 256^2} \right]$$

NC = 1

For Picture name = "Lena.bmp" CNN = 0.016

SNR = 23.64

relation a signal noise

PSNR = 30.846

peak relation a signal noise

$$KC := \frac{8}{ENTR}$$

KC = 8.259

**რეკურსიული (ტალღური) ალგორითმი
რეალიზაციის პროგრამა**

```

Insert Image name -- name := "LENA.bmp" <-----
                                                    LIST of IMAGE's
                                                    Ships.bmp
                                                    School.bmp
                                                    Rose.bmp
                                                    LENA.bmp

IM := READBMP(name)
rows(IM) = 256
cols(IM) = 256
k := 0 ..  $\frac{\text{rows(IM)}}{2^{\text{in}}} - 1$ 
l := 0 ..  $\frac{\text{cols(IM)}}{2^{\text{in}}} - 1$ 
 $\frac{\text{cols(IM)}}{2^{\text{in}}} - 1 = 127$ 
in := 1 <-----
1 - I Level
2 - II Level
3 - III Level

rows(IM) · cols(IM) = 65536    pix
rows(IM) · cols(IM) · 8 = 524288 bit
    
```



IM

	0	1	2	3	4	5	6
0	137	136	133	136	138	134	134
1	137	136	133	136	138	134	134
2	138	133	134	134	136	132	130
3	133	133	133	130	134	133	128
4	129	133	130	130	133	131	132
5	131	133	130	122	132	131	130
6	131	130	130	130	132	131	...

$$\begin{aligned}
 b_{i,j}^1 &= (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1}) / 4, \\
 b_{i,j}^2 &= (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1}) / 4, \\
 b_{i,j}^3 &= (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1}) / 4, \\
 b_{i,j}^4 &= (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1}) / 4.
 \end{aligned}$$

$$BB1_{k,l} := \text{round}\left[0.25 \cdot (IM_{2 \cdot k, 2 \cdot l} + IM_{2 \cdot k+1, 2 \cdot l} + IM_{2 \cdot k, 2 \cdot l+1} + IM_{2 \cdot k+1, 2 \cdot l+1})\right]$$

$$BH1_{k,l} := \text{round}\left[0.25 \cdot (IM_{2 \cdot k, 2 \cdot l} + IM_{2 \cdot k+1, 2 \cdot l} - IM_{2 \cdot k, 2 \cdot l+1} - IM_{2 \cdot k+1, 2 \cdot l+1})\right]$$

$$HB1_{k,l} := \text{round}\left[0.25 \cdot (IM_{2 \cdot k, 2 \cdot l} - IM_{2 \cdot k+1, 2 \cdot l} + IM_{2 \cdot k, 2 \cdot l+1} - IM_{2 \cdot k+1, 2 \cdot l+1})\right]$$

$$HH1_{k,l} := \text{round}\left[0.25 \cdot (IM_{2 \cdot k, 2 \cdot l} - IM_{2 \cdot k+1, 2 \cdot l} - IM_{2 \cdot k, 2 \cdot l+1} + IM_{2 \cdot k+1, 2 \cdot l+1})\right]$$

$$HMPAR(PC) := \text{histogram}(\text{trunc}(\max(PC) - \min(PC)), PC)$$



BB1

HB1



BH1

HH1

rows(BB1) = 128

cols(BB1) = 128

mean(HB1) = 0

mean(BH1) = -0

mean(HH1) = 0

min(HB1) = -52

max(HB1) = 63

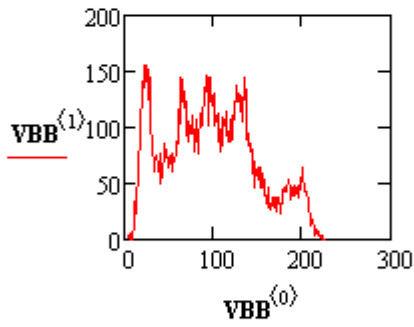
min(BH1) = -65

max(BH1) = 82

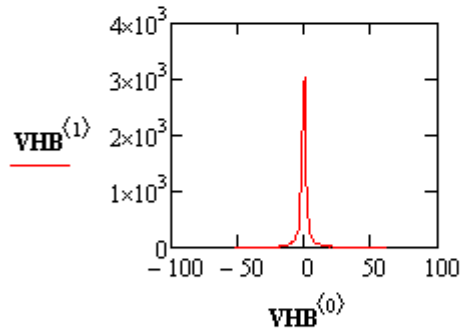
min(HH1) = -39

max(HH1) = 32

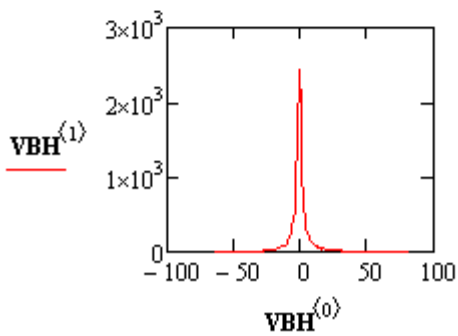
VBB := HMPAR(BB1)



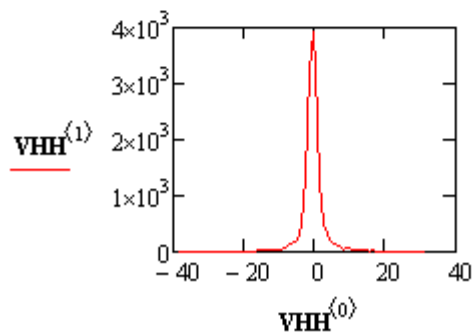
VHB := HMPAR(HB1)



VBH := HMPAR(BH1)



VHH := HMPAR(HH1)



$$\sum VHB^{(1)} = 16384$$

ENTR(IM) = 7.75

EBB := trunc(ENTR(VBB))

EBH := trunc(ENTR(VBH))

EHB := trunc(ENTR(VHB))

EHH := trunc(ENTR(VHH))

EBB = 7

EBH = 4

EHB = 3

EHH = 3

i := 0..1

j := 0..1

$$\text{ENTR(VPC)} := \left\{ \begin{array}{l} s \leftarrow \sum \text{VPC}^{(1)} \\ \text{SH} \leftarrow 0 \\ \text{for } i \in 0.. \text{rows(VPC)} - 1 \\ \quad \left\{ \begin{array}{l} \text{PL}_i \leftarrow \frac{\text{VPC}_{i,1}}{s} \\ \text{HE}_i \leftarrow -\text{PL}_i \cdot \log(\text{PL}_i, 2) \quad \text{if } \text{PL}_i \neq 0 \\ \text{HE}_i \leftarrow 0 \quad \text{otherwise} \\ \text{SH} \leftarrow \text{HE}_i + \text{SH} \end{array} \right. \\ \text{SH} \end{array} \right.$$

$$\text{JBB2}_{2 \cdot k+i, 2 \cdot l+j} := \text{if } [k = 127 \vee l = 127, \text{round} \left[\frac{(\text{BB1}_{k,1} + \text{BB1}_{k,1} + \text{BB1}_{k,1} + \text{BB1}_{k,1})}{4} \right], \\ \text{round} \left[\frac{(\text{BB1}_{k,1} + \text{BB1}_{k+i,1} + \text{BB1}_{k,l+j} + \text{BB1}_{k+i,l+j})}{4} \right]]$$

$$\text{JHB2}_{2 \cdot k+i, 2 \cdot l+j} := \text{if } [k = 127 \vee l = 127, \text{round} \left[\frac{(\text{HB1}_{k,1} + \text{HB1}_{k,1} - \text{HB1}_{k,1} - \text{HB1}_{k,1})}{4} \right], \\ \text{round} \left[\frac{(\text{HB1}_{k,1} + \text{HB1}_{k+i,1} - \text{HB1}_{k,l+j} - \text{HB1}_{k+i,l+j})}{4} \right]]$$

$$\text{JBH2}_{2 \cdot k+i, 2 \cdot l+j} := \text{if } [k = 127 \vee l = 127, \text{round} \left[\frac{(\text{BH1}_{k,1} - \text{BH1}_{k,1} + \text{BH1}_{k,1} - \text{BH1}_{k,1})}{4} \right], \\ \text{round} \left[\frac{(\text{BH1}_{k,1} - \text{BH1}_{k+i,1} + \text{BH1}_{k,l+j} - \text{BH1}_{k+i,l+j})}{4} \right]]$$

$$\text{JHH2}_{2 \cdot k+i, 2 \cdot l+j} := \text{if } [k = 127 \vee l = 127, \text{round} \left[\frac{(\text{HH1}_{k,1} - \text{HH1}_{k,1} - \text{HH1}_{k,1} + \text{HH1}_{k,1})}{4} \right], \\ \text{round} \left[\frac{(\text{HH1}_{k,1} - \text{HH1}_{k+i,1} - \text{HH1}_{k,l+j} + \text{HH1}_{k+i,l+j})}{4} \right]]$$

HHBB := JBB2 + JHB2 + JBH2 + JHH2

$$\text{ERROR} := \text{IM} - \text{HHBB}$$



HHBB

ERROR

$$\max(\text{HHBB}) = 228 \quad \max(\text{IM}) = 238 \quad \max(\text{ERROR}) = 116$$

$$\min(\text{HHBB}) = 5 \quad \min(\text{IM}) = 3 \quad \min(\text{ERROR}) = -97$$

$$\text{SNR}(\text{orig}, \text{file}) := 10 \cdot \log \left[\frac{\sum_{m=0}^{255} \sum_{n=0}^{255} (\text{orig}_{m,n})^2}{\sum_{m=0}^{255} \sum_{n=0}^{255} (\text{orig}_{m,n} - \text{file}_{m,n})^2} \right]$$

$$\text{PSNR}(\text{orig}, \text{file}) := -10 \cdot \log \left[\frac{\sum_{m=0}^{255} \sum_{n=0}^{255} (\text{orig}_{m,n} - \text{file}_{m,n})^2}{256^2 \cdot 256^2} \right]$$

$$\text{KC} := \frac{256 \cdot 256 \cdot 8}{128 \cdot 128 \cdot (\text{EBB} + \text{EBH} + \text{EHB} + \text{EHH})}$$

$$\text{KC} = 1.882$$

$$\text{SNR}(\text{IM}, \text{HHBB}) = 19.271 \quad \text{relation a signal noise}$$

$$\text{PSNR}(\text{IM}, \text{HHBB}) = 26.477 \quad \text{peak relation a signal noise}$$

$$\text{round}(\text{KC}) = 2$$

$$\text{HHBB1} := \text{JBB2} + \text{JBH2}$$

without JH12

$$\text{SNR}(\text{IM}, \text{HHBB1}) = 19.291 \quad \text{relation a signal noise (dB)}$$

$$\text{PSNR}(\text{IM}, \text{HHBB1}) = 26.496 \quad \text{peak relation a signal noise (dB)}$$

$$\text{KC} := \frac{256 \cdot 256 \cdot 8}{128 \cdot 128 \cdot (\text{EBB} + \text{EHB})}$$

$$\text{KC} = 3.2$$



HHBB1

1. Шульгин В.И. Основы теории передачи информации. Часть 1. Экономное кодирование. – Харьков, "ХАИ", 2003.
2. Левин Ю.С. Введение в теорию и технику радиотехнических систем. - М.: Радио и связь, 1986.
3. Зюко А.Г. Теория передачи сигналов. - М.: Сов. радио, 1972.
4. Радиотехнические системы /Под ред. Ю.М. Казаринова - М.: Сов. радио, 1968.
5. Чердынцев В.А. Радиотехнические системы. – Минск: Вышэйш. шк., 1988.
6. Пенин П.И. Системы передачи цифровой информации. - М.: Сов. радио, 1976.
7. Мордухович Л.Г., Степанов А.Г. Системы радиосвязи (курсовое проектирование). - М.: Радио и связь, 1987.
8. Системы радиосвязи /Под ред. Н.К. Калашникова - М.: Радио и связь, 1988.
9. Пышкин И. Н. Системы первичной радиосвязи. - М.: Радио и связь, 1988.
10. Тепляков И.П., Рощин Б.В. Радиосистемы передачи информации. - М.: Радио и связь, 1982.
11. Банкет В.Л., Дорофеев В.П. Цифровые методы в спутниковой связи. - М.: Радио и связь, 1988.
12. Кузьмин И.В. Основы теории информации и кодирования. – Минск: Вышэйш. шк., 1986.
13. Хемминг Р.В. Теория информации и теория кодирования. - М.: Радио и связь, 1983.
14. Ватолин Д.С. Алгоритмы сжатия изображений. – М: МГУ, 1999. http://graphics.cs.msu.su/library/our_publications/index.htm.
15. Ватолин Д.С. MPEG-стандарт ISO на видео в системах мультимедиа. E-mail: rois@red.com.ru.
16. Coding of moving pictures and associated audio. Committee Draft of Standart ISO11172: ISO/MPEG 90/176 Dec., 1990.
17. Le Gall D.A. MPEG: a video compression standart for multimedia applications. Communication of ACM. Volume 34. Number 4., April 1991.
18. JPEG digital compression and coding of continuus-tone still images. Draft ISO 10918, 1991.

იბეჭდება ავტორთა მიერ წარმოდგენილი სახით

გადაეცა წარმოებას 03.07.2009. ხელმოწერილია დასაბეჭდად
09.07.2009. ქალაქის ზომა 60X84 1/16. პირობითი ნაბეჭდი თაბახი 9.
ტირაჟი 100 ეგზ.

საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, თბილისი,
კოსტავას 77

