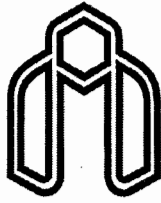


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده مهندسی برق

عنوان پایان نامه :

شمارش خودروها در یک بزرگراه با استفاده از

روشهای پردازش تصویر

این پایان نامه برای دریافت درجه کارشناسی ارشد

در مهندسی برق - گرایش کنترل می باشد.

دانشجو :

سید سعید میرشریفی

استاد راهنما:

دکتر حیدر طوسیان شاندیز

زمستان ۸۲

دستورالعمل اجرایی پایان نامه کارشناسی ارشد

در تحصیل علم بکوشید که فرا گرفتن آن حسنه و گفتگویش تسبیح و کاوش در آن جهاد و آموختن او به جاهل صدقه و نشرش موجب قربت است. زیرا علم راهنمای حلال است. طالب خود را به بهشت میکشد. در تنهایی مونس و در غربت یار و در سختی رهنما و در برابر دشمن حربه و برای دوست زیب و زیور است.

« پیامبر اکرم (ص) »

مقدمه :

به منظور ارتقاء و پرورش قدرت اعتماد به نفس ، ابداع و ابتکار و روح تتبع دانشجویان در فعالیتهای مستقل علمی ، پژوهشی و حل معضلات علمی و پژوهشی در راستای استقلال فرهنگی ، سیاسی، اقتصادی و خودکفایی کشور و نیز حصول اطمینان نسبی دانشگاه نسبت به آشنایی فارغ التحصیلان با روشهای پیشرفته تحقیق و آخرین دستاوردهای علمی ، لازم است دانشجو در یکی از جنبه های رشته تحصیلی خویش به انجام کار تحقیقی بپردازد و نتایج آن را به صورت پایان نامه ای مطابق مفاد این دستورالعمل ارائه نماید.

پایان نامه کارشناسی ارشد

۱- تعریف

ارائه پایان نامه کارشناسی ارشد آخرین مرحله تحصیلی دوره کارشناسی ارشد است که طی آن دانشجو موظف است در یک زمینه مربوط به رشته تحصیلی خود ، زیر نظر استاد راهنما به تتبع و تحقیق علمی (نظری و عملی) بپردازد. تحقیق میتواند به سه صورت نظری ، کاربردی و یا بنیادی باشد.

الف - تحقیقات نظری: به تحقیقاتی گفته می شود که صرفاً در مرحله ارائه نظریه بوده و اثبات و رد آن نیز در محدوده نظریات است و امکان کاربردی تجربی ندارد.

ب - تحقیقات کاربردی : به تحقیقاتی گفته می شود که با به کارگیری اصول و فرضیه های اساسی در یک زمینه و یا هدف خاص بتواند به طور مستقیم و یا غیر مستقیم مورد استفاده اعضای جامعه قرار بگیرد .

ج - تحقیقات بنیادی : به تحقیقاتی اطلاق می شود که سعی در اثبات یا نفی کامل یا جزئی یک فرضیه یا نظریه یا اصول اساسی علمی دارد.

ب - استاد راهنما به پیشنهاد دانشجو ، موافقت استاد و تأیید شورای تحصیلات تکمیلی دانشکده تعیین می‌شود.

ج - اساتید راهنما باید دارای مدرک دکتری با حداقل مرتبه استادیاری باشند.

د - در رشته های علوم انسانی و هنر ، علما ، محققین و متخصصین مربوطه با مدرک معادل استادیاری میتوانند راهنمایی پایان نامه ها را به عهده گیرند.

ه - وظایف استاد راهنما به شرح زیر است :

- دارا بودن مسئولیت اصلی و مستقیم در هدایت پایان نامه

- همکاری با دانشجو در تکمیل فرم مخصوص « طرح تحقیق پایان نامه »

- هدایت علمی دانشجو در کلیه مراحل اجرای پایان نامه

- تشکیل جلسات مستمر با دانشجو به منظور بررسی پیشرفت کار پایان نامه

- صدور مجوز دفاع از پایان نامه

- شرکت در جلسه دفاعیه پایان نامه بعنوان یک عضو از هیات داوران

۴- استاد مشاور

الف - استاد مشاور به پیشنهاد استاد راهنما و پس از تأیید شورای تحصیلات تکمیلی دانشکده از بین اعضاء هیات علمی یا متخصصان و محققان مربوط انتخاب تعیین می‌شود.

ه - وظایف استاد مشاور به شرح زیر است :

- دارا بودن مسئولیت ثانوی در هدایت پایان نامه

- همکاری فعال و مستقیم با دانشجو در مراحل اجرای پایان نامه

- شرکت در جلسه دفاعیه پایان نامه بعنوان یک عضو از هیات داوران

تبصره - کلیه نظرات استاد مشاور در چارچوب پایان نامه می بایستی با توافق استاد راهنما اعمال گردد.

۵ - دفاع از پایان نامه

دفاع از پایان نامه پس از تدوین آن توسط دانشجو و تأیید کیفیت علمی و صحت مطالب آن

از طرف استاد راهنما، مبنی بر قابل دفاع بودن پایان نامه ، در حضور هیات داوران صورت می‌گیرد.

الف - دفاع از پایان نامه و پذیرفته شدن آن به منظور اخذ دانشنامه کارشناسی ارشد در تمامی رشته های تحصیلی دانشگاه صنعتی شاهرود الزامی است.

ب) ترکیب هیات داوران به شرح زیر است :

۱- استاد راهنما

۲- استاد مشاور

دستورالعمل نگارش پایان نامه کارشناسی ارشد

نحوه نگارش پایان نامه

مقدمه :

یکی از اهداف ارائه پایان نامه ، آشنایی دانشجویان گرامی با نحوه نگارش و تنظیم مطالب یک موضوع علمی است. در اینجا تمام نکات لازم را نمی توان بیان کرد، اما برای ایجاد هماهنگی بین پایان نامه ها ، لازم است دانشجویان نکات زیر را در تهیه پایان نامه رعایت نمایند.

یک پایان نامه به ترتیب شامل بخشهای زیر می باشد:

- بسم ...

- تاییدیه اعضای هیات داوران حاضر در جلسه دفاع (طبق نمونه فرم پیوست - * فقط برای پایان نامه‌ای که به زبان انگلیسی تدوین شده انگلیسی آن الزامی است) .

- درج آئین نامه چاپ - پایان نامه

- صفحه عنوان (طبق فرم پیوست در راهنمای پژوهشی دانشجویان)

- تقدیم (Dedication) (در یک صفحه) در صورت لزوم

- تشکر و قدردانی (Acknowledgements) (در یک صفحه)

- چکیده (Abstract) (حداکثر ۳۰۰ کلمه شامل کلمات کلیدی ، ترجیحاً در یک صفحه)

- فهرست مطالب (Table of Contents)

- فهرست نمودارها، عکسها و نقشه ها (Lists of Figures, Pictures and Maps) (در صورت نیاز)

- فهرست جداول (List of tables) (در صورت نیاز)

- فهرست علائم و نشانه ها (Abbreviations) (در صورت نیاز)

- متن اصلی پایان نامه که شامل قسمتهای زیر می باشد: (گروههای تخصصی با توجه به نیاز خود میتوانند در بند ۳ جزئیات بیشتری را اعمال کنند).

۱- مقدمه (Introduction)

۲- مروری بر مطالعات انجام شده (Literature Review)

۳- فصلهای مربوط به کار اصل تحقیق (در صورت نیاز)

۴- بحث (Discussion) نتایج (Results) ، و پیشنهادات (Suggestions)

۵- فهرست منابع و ماخذ (References)

۶- ضمائم (در صورت نیاز)

۷- چکیده به انگلیسی (الزامی است) : چکیده انگلیسی باید تا حد امکان ترجمه چکیده فارسی و شامل کلمات کلیدی باشد .

۸- صفحه عنوان انگلیسی (الزامی است - طبق فرم پیوست)

- اهداف و موارد بررسی شده باید مورد توجه قرار گیرد.

- تاکید روی اطلاعات تازه (یافته ها) و اصطلاحات جدید یا نظریه ها ، فرضیه ها ، نتایج و پیشنهادات متمرکز شود.

- اگر در پایان نامه روشی نوینی برای اولین بار ارائه می شود و تا به حال معمول نبوده است ، با جزئیات بیشتری ذکر شود.

شایان ذکر است چکیده باید حتماً به تائید استاد راهنما رسیده باشد.

۵ - فهرستها

بعد از چکیده پایان نامه به ترتیب فهرست مطالب ، فهرست علائم و نشانه ها ، فهرست جداول ، فهرست اشکال و سایر فهرستها قرار می گیرند. در فهرست مطالب صفحات اولیه پایان نامه (تقدیر ، تقدیم ، چکیده و ...) با شماره گذاری حرفی ذکر شود. فهرست مطالب شامل فهرستهای جداول ، اشکال ، علائم اختصاری ، عناوین فصول و بخشها و زیر بخشهای هر فصل پایان نامه با ذکر شروع شماره صفحه و عنوان مربوطه می باشد. عناوین ذکر شده در فهرست مطالب باید عیناً با عناوین خود متن مطابقت کند : (برای مثال شماره گذاری ، نقطه گذاری ، با حروف بزرگ نوشتن) . بعبارت دیگر فهرست عناوین باید کاملاً همانند عناوین در داخل متن اصلی باشد.

- در فهرست ها بین دو عنوان پشت سر هم باید فاصله بیشتر باشد که دو عنوان را از یکدیگر متمایز کند ولی برای یک عنوان که بیش از یک خط است باید فاصله معمولی بکار برده شود و شماره صفحه در خط دوم ذکر شود.

- برای جلوگیری از خطای چشم بین عنوان و شماره صفحه آن در فهرست مطالب نقطه چین تایپ شود.

- فهرستهای مطالب ، علائم و نشانه ها ، جداول ، اشکال و سایر فهرستها با حروف ابجد شماره گذاری شود.

- بترتیب مراجع ، ضمامم (در صورت نیاز) و چکیده انگلیسی که بدنبال متن اصلی پایان نامه / رساله قرار میگیرند بصورت تیتیری مجزا از فصول دیگر پایان نامه در فهرست مطالب تایپ میشوند.

۶ - مقدمه

اولین فصل از ساختار اصلی پایان نامه است . هدف از نوشتن مقدمه این است که زمینه اطلاعاتی لازم برای خواننده فراهم آید. در طول مقدمه باید سعی شود موضوع تحقیق با زبانی روشن ، ساده و بطور عمیق و جهت یافته به خواننده معرفی شود. مقدمه باید خواننده را مجذوب و اهمیت موضوع تحقیق را آشکار سازد. در مقدمه یا فصل ۲ با ارائه سوابق ، شواهد تحقیقی ، اطلاعات موجود و مطالعات انجام شده (با ذکر منبع) به روش منظم ، منطقی و هدف دار خواننده را جهت داد و به سوی راه حل مورد نظر هدایت کرد. بالاخره شاید اشاره به نتایج کلی در مقدمه خالی از فایده نباشد. همچنین در مقدمه میتوان به فصول اصلی مربوط به تحقیق حسب نیاز موضوع و نظر ساتاد راهنما اشاره کرد.

مقدمه مناسبترین جا برای ارائه اختصارات و بعضی توضیحات کلی است توضیحاتی که شاید نتوان در مباحث دیگر در مورد آنها توضیح داد.

۷- فصول اصلی مربوط به تحقیق حسب نیاز موضوع و نظر استاد راهنما تنظیم گردد.

شماره گذاری صفحات تا قبل از فصل مقدمه با اعداد به صورت حرفی (یک ، دو ، سه و ...) انجام میشود.
شماره گذاری با اعداد از اولین صفحه « مقدمه » آغاز میشود و تا آخرین صفحه ادامه می یابد. شماره صفحات باید در وسط و چسبیده به قسمت فوقانی حاشیه پایین صفحه درج شود. کلیه صفحات متن اصلی پایان نامه باید شماره گذاری شوند.

ب - شماره گذاری موضوعات

موضوعات اصلی پایان نامه به چند فصل و هر فصل به چند « بخش » تقسیم میشود. صفحه اول هر فصل باید از خط پنجم شروع شود.

هر یک از بخشهای هر فصل با دو شماره که با خط فاصل از یکدیگر جدا شده اند، مشخص میشود. عدد سمت راست بیانگر شماره فصل و عدد سمت چپ شماره ترتیب بخش مورد نظر است . در صورتیکه هر بخش دارای زیر بخش باشد، شماره هر زیر بخش در سمت چپ عدد فوق قرار میگیرد. به عنوان مثال اگر برنامه ریزی پویا دومین زیر بخش از بخش چهارم فصل سوم باشد، بصورت : ۳ - ۴ - ۲ برنامه ریزی پویا نوشته خواهد شد.

ج - شماره گذاری اشکال و جداول

شماره گذاری اشکال و جداول نیز از همان قانون بالا تبعیت می کند بعنوان مثال هفتمین شکل در فصل دوم به صورت (شکل ۲ - ۷ - نمودار تغییرات ...) نوشته میشود. دقت شود شماره گذاری جداول و اشکال مستقل از یکدیگر است.

د - شماره گذاری ضمائم

ضمائم پایان نامه با حروف ابجد نامگذاری میشوند. (ضمیمه الف یا ضمیمه ب) و شماره گذاری صفحات ضمائم به همان ترتیب متن اصلی پایان نامه می باشد. اشکال و جداول موجود در هر ضمیمه نیز با توجه به ضمیمه مربوطه شماره گذاری خواهد شد، به عنوان مثال دومین جدول در ضمیمه الف بصورت زیر نوشته می شود: (جدول الف - ۲)

ه - شماره گذاری روابط و فرمولها

هر رابطه که در متن باشد با دو شماره که با خط فاصله از یکدیگر جدا میشوند مشخص می شود. عدد سمت راست بیانگر شماره فصل و عدد سمت چپ شماره رابطه مورد نظر است . مثلاً دهمین رابطه در فصل هفتم به صورت (۷ - ۱۰) نوشته میشود. روابطی که در ضمائم آمده است با توجه به حروف ابجد آن ضمیمه ، شماره گذاری میشود. مانند رابطه (الف - ۲)

طرح تحقیق پایان نامه کارشناسی ارشد

دانشکده: برق گروه: برق

این قسمت توسط دفتر تحصیلات تکمیلی دانشکده پرمیشود

شماره:
تاریخ درخواست:
پیوست:

مشخصات دانشجو:	
نام خانوادگی: میرشریفی	نام: سید سعید
دانشکده: برق	شماره دانشجویی: ۸۱۳۶۱۰۴
رشته: برق	گرایش: کنترل

آدرس دانشجو: تهران - اتوبان شهید محلاتی - خ عزف - روبروی اداره برق - کوچه فرشچی نصر - پلاک ۱۸
تلفن: ۲۷۰۸۴۶۴

مشخصات استاد یا اساتید راهنما					
ردیف	نام و نام خانوادگی	تخصص اصلی	آخرین مدرک تحصیلی	رتبه دانشگاهی	محل خدمت
۱	حیدر طوسیان شاندیز	ابزار دقیق	دکتری	استادیار	دانشگاه صنعتی شاهرود
۲	_____	_____	_____	_____	_____

عنوان پایان نامه
فارسی: شمارش خودروها در یک بزرگراه با استفاده از روشهای پردازش تصویر
لاتین: Counting Vehicles In High Way With Using Image Processing Methods

*توجه: این فرم باید با نظارت و هدایت استاد راهنما پر شود.

(فرم شماره ۱)

۱	اطلاعات مربوط به دانشجو:	
نام و نام خانوادگی: سید سعید میرشریفی		شماره دانشجویی: ۸۱۳۶۱۰۴
دانشکده: برق	رشته تحصیلی: برق	گرایش: کنترل
سال ورود: بورسیه از:	مأمور به تحصیل از:	آزاد <input checked="" type="checkbox"/> نوبت اول <input type="checkbox"/> نوبت دوم <input type="checkbox"/>
آدرس و تلفن در شهر محل تحصیل: شاهرود - دانشگاه صنعتی شاهرود - خوابگاه ۱ - اتاق ۲۸		
آدرس و تلفن شهرستان محل سکونت: تهران - اتوبان شهید محلاتی - خ عارف - روبروی اداره برق - کوچه فرشچی نصر - پلاک ۱۸ . تلفن ۳۷۰۸۴۶۴		

۲	اطلاعات مربوط به استاد راهنما(استاد راهنمای اول)	
نام و نام خانوادگی: حیدر طوسیان شانديز		تخصص اصلی: ابزار دقیق
آخرین مدرک تحصیلی: دکتری		سال اخذ مدرک: ۱۳۷۹
سنوات تدریس در دوره کارشناسی ارشد: دو سال تمام		سنوات تدریس در دکترا: -----
تعداد پایان نامه های راهنمایی شده: در شرف اتمام		تعداد پایان نامه های در دست راهنمایی: -----
کارشناسی ارشد: ۲	دکتری: -----	کارشناسی ارشد: ----- دکتری: -----
در دانشگاه صنعتی شاهرود: بله	در سایر دانشگاهها:	در دانشگاه صنعتی شاهرود: در سایر دانشگاهها:
سمت فعلی: عضو هیئت علمی - مدیر مرکز رشد فناوری		محل خدمت: دانشگاه صنعتی شاهرود
آدرس و تلفن: دانشگاه صنعتی شاهرود		۰۹۱۱۲۷۳۳۶۳۳

۳	اطلاعات مربوط به استاد راهنما(استاد راهنمای دوم)	
نام و نام خانوادگی: -----		تخصص اصلی: -----
آخرین مدرک تحصیلی: -----		سال اخذ مدرک: -----
سنوات تدریس در دوره کارشناسی ارشد: -----		سنوات تدریس در دکترا: -----
تعداد پایان نامه های راهنمایی شده: -----		تعداد پایان نامه های در دست راهنمایی: -----
کارشناسی ارشد: -----	دکتری: -----	کارشناسی ارشد: ----- دکتری: -----
در دانشگاه صنعتی شاهرود: -----	در سایر دانشگاهها:	در دانشگاه صنعتی شاهرود: در سایر دانشگاهها:
سمت فعلی: -----		محل خدمت: -----
آدرس و تلفن: -----		

۴	اطلاعات مربوط به اساتید مشاور:		
ردیف	نام و نام خانوادگی	تخصص اصلی	آخرین مدرک تحصیلی
			رتبه دانشگاهی

				۱
--	--	--	--	---

سوابق آموزشی، پژوهشی و اجرایی :

آدرس و تلفن:

رتبه دانشگاهی	مدرک	آخرین تحصیلی	تخصص اصلی	نام و نام خانوادگی
۲				

سوابق آموزشی، پژوهشی و اجرایی :

آدرس و تلفن:

	اطلاعات مربوط به پایان نامه
--	-----------------------------

الف - عنوان پایان نامه

	شمارش خودروها در یک بزرگراه با استفاده از روشهای پردازش تصویر
	الف - ۱ - فارسی
	الف - ۲ - لاتین

ب - واژگان کلیدی

	ب - ۱ - فارسی : پردازش تصویر دیجیتال - کنترل ترافیک - ابزار - شمارش خودرو
	ب - ۲ - لاتین : Counting Vehicle-Instrumentation - Traffic Control - Digital Image Processing

	۱-۵	نوع تحقیق	<input type="checkbox"/> بنیادی	<input type="checkbox"/> نظری	<input checked="" type="checkbox"/> کاربردی	<input type="checkbox"/> توسعه ای
--	-----	-----------	---------------------------------	-------------------------------	---	-----------------------------------

	۲-۵	تعداد واحد پایان نامه	۶ :	۳-۵	مدت اجرا :	یک سال
--	-----	-----------------------	-----	-----	------------	--------

	۴-۵
	<p>تعریف مساله و بیان سوالهای اصلی تحقیق:</p> <p>مساله کنترل ترافیک یکی از مهمترین مسایل روز است ، که برای حل مشکل ترافیک و کنترل آن راههای متعددی پیشنهاد شده است که هر یک معایب و مزایای خاص خودشان را دارند. راه حل ما در این پایان نامه مبتنی بر کنترل ترافیک بوسیله شمارش خودرو در هر بزرگراه است .</p> <p>در این پایان نامه ابتدا تصاویری از فیلمهای دریافتی از یک خیابان نمونه گیری شده ، دیجیتال می شوند(که همه این اعمال با یک دور بین دیجیتال تقریبا بدون زمان بری ممکن می گردند). سپس با ورود آنها به کامپیوتر و انجام عملیاتهای پردازش تصویر(Thersholding بین سطوح خاکستری و نیز استفاده از داده های آماری مربوط به محل نصب دوربین) و برنامه ارایه شده تعداد وسایل نقلیه تخمین زده می شوند و با توجه به ظرفیت بزرگراه پیغامی مبتنی بر ازدحام (بالا- متوسط - ضعیف) ظاهر خواهد شد. از این برنامه می توان برای کنترل Timer چراغهای راهنمایی استفاده کرد .</p>

۵-۵	سابقه و ضرورت انجام تحقیق :
<p>در حال حاضر همان طور که گفته شد چندین روش برای انجام چنین کنترلی بر ترافیک وجود دارد، ولی همه آنها مشکلاتی نظیر : طولانی بودن زمان اجرای برنامه یا نیازمند بودن به وسایل و ابزارهای جانبی گران قیمت و ... را دارند ، که انجام چنین کاری را خیلی کند یا غیر مقرون به صرفه می کند . چون هر یک از این روشها از چندین تابع و عملگر پردازش تصویر استفاده می کند ، ولی هیچیک از آنها تا بحال از داده های آماری مربوط به دوربین (ارتفاع محل نصب ، میدان دید ،) استفاده نکرده اند ، که این خود نشانه جدید بودن این ایده و روش است .</p> <p>ما در این متد جدید (استفاده از داده های آماری و Thersholding) نه تنها مشکل زمان اجرا را کم رنگ می - کنیم ، بلکه با قیمت کمتری (بعلت استفاده نکردن از پردازشگرهایی با سرعت بسیار بالا) توانایی انجام این کار (شمارش خودرو و در نهایت کنترل ترافیک) را داریم .</p> <p>از ضرورت های انجام این نوع تحقیقات اینست که ظرفیت خیابانها محدود بوده و میزان تولید و مصرف ماشینها رو به افزایش است و اگر با انجام این روشها ترافیک ار کنترل نکنیم در چند سال نه چندان دور دچار معزلی شدیدا غیر قابل جبران می شویم .</p>	

۶-۵	فرضیه ها (هر فرضیه بصورت یک جمله خبری نوشته شود)
<p>فرض می کنیم که یک تصویر خالی از ماشینها برای کنترل هر خیابان داریم . تصاویر Scan شده یا دریافتی از دوربین دیجیتال باید دارای ابعاد ۲۵۶x۲۵۶ یا بیشتر باشند و ۸ بیت خانه حافظه داشته باشند (۲۵۶ سطح خاکستری) تصاویر از بالا به صورت عمودی قسمتی از جاده را پوشش می دهند .</p>	

۷-۵	هدفها
<p>تشخیص تغییرات در تصاویر دریافتی (از نظر وجود و یا عدم وجود خودروها) پردازش تصاویر دریافتی و شمارش تعداد وسایل نقلیه در یک تصویر . با استفاده از شمارش تعداد آنها پیغامی مبتنی بر میزان شلوغی بزرگراه را می دهیم . سعی بر On time & On line (بلا درنگ کردن) عملیات داریم .</p>	

۸-۵	چه کاربردهایی از انجام تحقیق متصور است
<p>کنترل ترافیک خودرو ها و ایجاد تعادل و پخش ترافیک بزرگراهها و جلوگیری از تجمع خودروها در یک محل و خالی ماندن محلهای دیگر است . با توجه به نوع متد بکار رفته از آن برای شمارش و کنترل جمعیت در پیاده رو ها نیز استفاده می شود .</p>	

استفاده کنندگان از نتیجه پایان نامه	۹-۵
<p>۱- مراکز کنترل ترافیک برای کنترل سریعتر و بهتر</p> <p>۲- کسانی که به هر جهت می خواهند تعداد شیء را در تصویر به طور مرتب حساب کنند .</p> <p>۳- ارایه در مجلات علمی</p>	

جنبه جدید و نوآوری تحقیق	۱۰-۵
<p>این روش از آنجایی که در روشهای ماقبل از این روش از داده های آماری محل نصب دوربین استفاده نمی شد و بجای آن از الگوریتمهای پیچیده ای بهره می بردند ، جدید و نو بنظر می رسد .</p> <p>این قسمت توسط استاد راهنما پز شود.</p>	

روش انجام تحقیق	۱۱-۵
<p>پس از انجام مطالعات مقدماتی و تعیین تاریخچه کارهای انجام شده تا کنون ، مرحله اجرایی و نوشتن برنامه کامپیوتری لازم آغاز می شود و سعی می شود که نتایج حاصله در حد امکان کاربردی باشند تا در مراکز کنترل ترافیک بکار روند .</p>	

روش و ابزار گردآوری اطلاعات	۱۲-۵
<p>جدید ترین کتب کاربردی منتشر شده در زمینه پردازش تصویر و ترافیک اینترنت و مقالات معتبر در <i>IEEE</i> مصاحبه با اهل فن مراجع و کتب مرکز تحقیقات کنترل ترافیک و کتب دانشگاهی</p>	

جامعه آماری و تعداد نمونه (در صورت لزوم)	۱۳-۵
<p>در صورت لزوم سعی شده است که اطلاعات را خودمان و نیز با کمک جنبی مراکز تحقیقات کنترل ترافیک جمع آوری کنیم .</p>	

روش نمونه گیری (در صورت لزوم)	۱۴-۵
<p>-----</p>	

هزینه های پرسنلی (برای مواردی که در حوزه تخصص و مهارت دانشجو قرار ندارد)	۲-۶
--	-----

نوع مسئولیت	تعداد پرسنل	کل ساعات کار	دستمزد در ساعت	جمع کل به ریال
		جمع کل به ریال		

هزینه های مسافرت	۳-۶
------------------	-----

مقصد	تعداد سفر	نوع وسیله	هزینه هر سفر	جمع به ریال
		جمع کل به ریال		

ردیف	نوع هزینه	مبلغ به ریال
۱	مواد و وسایل	
۲	پرسنلی	
۳	مسافرت	
۴	متفرقه	
۵	جمع کل	
۴-۶	جمع کل هزینه ها	

محل امضا		۷
تاریخ	امضا	نام و نام خانوادگی دانشجو سید محمد نوری
تاریخ	امضا	نام و نام خانوادگی استاد راهنما (اول)
تاریخ	امضا	نام و نام خانوادگی استاد راهنما (دوم)
تاریخ	امضا	نام و نام خانوادگی استاد مشاور
تاریخ	امضا	نام و نام خانوادگی استاد مشاور

نظریه شورای تخصصی گروه

طرح پایان نامه کارشناسی ارشد آقای / خانم

، گرایش پترولوژی تحت عنوان:

به راهنمایی آقای

مطرح و مورد موافقت قرار گرفت

قرار نگرفت

دانشجوی کارشناسی ارشد

شورای گروه

در جلسه مورخ

امضاء مدیر گروه

علت عدم موافقت :

نظریه شورای تحصیلات تکمیلی دانشکده

طرح پایان نامه کارشناسی ارشد آقای / خانم

، گرایش تحت عنوان:

به راهنمایی آقای

که در جلسه مورخ

تصویب رسیده بود، در جلسه مورخ

مطرح و به اتفاق آراء به تصویب رسید.

دانشجوی کارشناسی ارشد

شورای گروه

شورای تحصیلات تکمیلی دانشکده

ردیف	نام و نام خانوادگی اعضای جلسه	امضاء
۱		
۲		
۳		
۴		
۵		
۶		

تقدیم به پدر و مادر عزیزم که پس از خداوند
بزرگ ، بزرگترین تکیه گاه و یاری دهنده من
بوده و هستند .

سعید

تشکر و قدردانی از استاد بزرگوارم
جناب آقای دکتر حیدر طوسی‌ان که با
کمک‌های بی‌دریغ خود، برای همیشه
من را رهین منت خویش ساختند.

خلاصه

این پایان نامه شامل دو بخش است . بخش اول در مورد پایه ها و فواید و کاربرد ها و طرز عملکرد عملگرهای پردازش تصویر دیجیتال بحث می کند . این بخش با نمایش دو برنامه در مورد این موضوعات کامل می شود ، ضمیمه (۱ و ۲). این بخش شامل سه فصل :

۱- بازسازی تصاویر نویزی

۲- آشنایی با نرم افزار پردازش تصویر

۳- تبدیل ها و فشرده سازی تصاویر

می باشد.

در بخش دوم ، ما یک سیستم بینایی کامپیوتری را برای شمارش ماشینهای روی جاده ها و خیابان ها مبتنی بر روش های پردازش تصویر دیجیتال توصیف کردیم . تصاویر ترافیکی توسط یک دوربین فیلم برداری گرفته می شوند و بعد از دیجیتال شدن وارد کامپیوتر می گردند. الگوریتم شمارش بر اساس مقایسه بین یک مرجع محاسبه شده بدون ماشین ها و تصویر جاری از خطوط ترافیکی بنا شده است . امتحان این روش تحت شرایط واقعی رضایت بخش بوده است و با این روش بالای ۹۰٪ درستی کار داریم .

کلمات کلیدی : پردازش تصویر دیجیتال ، کنترل ترافیک ، ابزار ، شمارش وسایل نقلیه .

عنوان مطالب	صفحه
تقدیم	یک
تشکر و قدردانی	دو
خلاصه فارسی	سه
فهرست مطالب	چهار

بخش اول - پردازش تصویر

فصل اول - بازسازی تصاویر نویزی

۱-۱- مقدمه	۱
۱-۲- علم بازیابی تصویر	۱
۱-۳- معیار بیز	۳
۱-۴- تبدیل فوریه	۶
۱-۵- تخمین تابع تبدیل	۷
۱-۵-۱- تبدیل فوریه CROSS-CORRELATION	۸
۱-۵-۲- تبدیل فوریه AUTO-CORRELATION	۸

فصل دوم - آشنایی با نرم افزار پردازش تصویر

۲-۱- مقدمه	۹
۲-۲- شروع عملیات پردازش تصویر	۹
۲-۳- برنامه جامع و کاربردی پردازش تصویر	۱۶
۲-۴- نتیجه	۱۶

صفحه	عنوان مطالب
	فصل سوم - تبدیل ها و فشرده سازی تصاویر
۱۷	۱-۳- فشرده سازی تصویر
۱۷	۲-۳- چکیده فصل
۱۸	۳-۳- هارتلی
۱۸	۱-۳-۳- مزایای تبدیل هارتلی
۱۹	۴-۳- ملزومات سیستم های تصویری
۲۰	۵-۳- معرفی تبدیل هارتلی
۲۰	۱-۵-۳- تاریخچه هارتلی
۲۱	۶-۳- توابع مختلط
۲۲	۷-۳- تبدیل هارتلی
۲۳	۱-۷-۳- ضرب در حوزه زمان
۲۳	۲-۷-۳- کانولوشن در حوزه زمان
۲۴	۳-۷-۳- روابط حوزه زمان و فرکانس فوریه و هارتلی
۲۵	۸-۳- توابع مختلط (۲)
۲۶	۹-۳- تبدیل (CAS-CAS)T
۲۷	۱۰-۳- تبدیل هارتلی گسسته DHT
۲۸	۱۱-۳- تبدیل هارتلی سریع FHT
۳۰	۱۲-۳- فشرده سازی تصویر
۳۱	۱۳-۳- تاریخچه فشرده سازی اطلاعات

عنوان مطالب	صفحه
۱۴-۳- ساختار کلی سیستم های فشرده سازی اطلاعات.....	۳۲
۱۵-۳- دی کدینگ.....	۳۲
۱۶-۳- روشهای فشرده سازی اطلاعات.....	۳۳
۱۷-۳- خلاصه روشهای فشرده سازی	۳۴
۱۸-۳- کدینگ طول گام Run-Length Coding.....	۳۵
۱۹-۳- کدینگ مدلاسیون پهنای پالس تفاضلی DPCM.....	۳۵
۲۰-۳- روش چندی سازی برداری Vector Quantization.....	۳۵
۲۱-۳- کدینگ تبدیل.....	۳۶
۱-۲۱-۳- خواص این تبدیل ها.....	۳۶
۲۲-۳- خواص یک تبدیل برای فشرده سازی مؤثر.....	۳۷
۱-۲۲-۳- از بین بردن همبستگی ها	۳۷
۲-۲۲-۳- خطی بودن	۳۸
۳-۲۲-۳- اورتوگونالیتهی.....	۳۸
۴-۲۲-۳- عملکرد سریع	۳۸
۲۳-۳- انتخاب تبدیل مناسب.....	۳۹
۲۴-۳- ساختار الگوریتمهای یک بعدی و دو بعدی.....	۴۱
۲۵-۳- تبدیل کسینوس گسسته DCT.....	۴۲
۲۶-۳- کدینگ تلفیق.....	۴۲
۲۷-۳- تبدیل دو بعدی هارتلی.....	۴۳

صفحه	عنوان مطالب
۴۴	۳-۲۸- چگونگی اعمال یک تبدیل به تصویر.....
۴۴	۳-۲۹- چگونگی حذف ضرایب در تصویر تبدیل یافته برای فشرده سازی تصویر.....
۴۵	۳-۳۰- خطای باز سازی.....
۴۵	۳-۳۱- کاربردهای دیگر تبدیل هارتلی.....

بخش دوم - کنترل ترافیک

فصل چهارم - مرور کارهای انجام شده تا کنون

۴۶	۴-۱- مقدمه.....
۴۸	۴-۲- کنترل ترافیک به روش موزائیک (Mosaik Method).....
۵۰	۴-۳- کنترل ترافیک به روش تخمین سرعت وسایل نقلیه.....
۵۱	۴-۳-۱- بحث.....
۵۱	۴-۴- روشهای دیگر.....
۵۴	۴-۵- روش Tracking & Speed estimating.....
۵۵	۴-۵-۱- روش آشکار سازی.....
۵۶	۴-۵-۲- روش محاسبه سرعت.....

فصل پنجم - معرفی روش جدید و پیشنهادی پایان نامه

۵۷	۵-۱- معرفی روش جدید و پیشنهادی پایان نامه.....
۵۸	۵-۲- سیستم اندازه گیری میزان ترافیک.....

صفحه	عنوان مطالب
۵۹	۵-۳- بیان مشروح روش
۶۱	۵-۴- قانون محاسبه فاصله خطوط
۶۱	۵-۵- خطای نسبی
۶۱	۵-۶- نکات برتری این روش
۶۲	۵-۷- بررسی حالات مختلف
۶۷	۵-۸- طرز پیاده سازی الگوریتم و نوشتن برنامه
۶۷	۵-۹- نتیجه بحث
۶۸	۵-۱۰- روش تعیین تعداد خطوط
۶۹	۵-۱۱- مزایا و کاربردهای روش
۷۰	۵-۱۱-۱- مزایای دیگر
۷۱	۵-۱۲- سیستم ترافیک
۷۱	۵-۱۳- طرز استفاده از نتایج حاصله
۷۲	۵-۱۴- مثال دیگر
۷۳	۵-۱۵- حل مثالهای اولیه با تصاویر واقعی از بزرگراه ها با این روش
۷۴	۵-۱۶- طرز تسخیر تصاویر
۸۰	۵-۱۷- نتایج
۸۱	۵-۱۸- حل مثالهای نهایی با تصاویر واقعی از بزرگراه ها با این روش

نتایج و پیشنهادات

نتیجه گیری..... ۹۵

پیشنهادات..... ۹۷

ضمائم

ضمیمه (۱)- برنامه جامع و کاربردی پردازش تصویر..... ۱۰۰

ضمیمه (۲)- فشرده سازی تصاویر به روش DCT با تعداد متغیر ضرایب تبدیل..... ۱۰۸

ضمیمه (۳)- کدهای برنامه اثبات درستی منطق جدید در محیط Visual Basic 6.0..... ۱۱۲

ضمیمه (۴)- کدهای برنامه شمارش ماشین‌ها در یک خط به همراه خروجی..... ۱۱۴

ضمیمه (۵)- کدهای برنامه شمارش کل ماشین‌ها در یک تصویر به همراه یک مثال از دیگر کاربردهای این روش..... ۱۲۱

ضمیمه (۶)- برنامه حل مثالهای اولیه با تصاویر واقعی از بزرگراه..... ۱۲۷

فهرست منابع و مآخذ

فهرست منابع و مآخذ..... ۱۳۱

واژه نامه ها

واژه نامه فارسی به انگلیسی..... ۱۳۴

واژه نامه انگلیسی به فارسی..... ۱۳۷

Abstract..... ۱۳۸

بخش اول

پردازش تصویر

فصل (۱)

باز سازی تصاویر نویزی

۱-۱- مقدمه

تاریخچه پردازش تصویر به سال ۱۹۲۰ بر می گردد ، اما بحث جدی در این مورد از سال ۱۹۶۰ آغاز شد. تصویری که از یک فیلم عکاسی حاصل می شود ، هرگز یک شکل دقیق از مشخصات محلی اشیاء نیست . محدود بودن قدرت تفکیک محلی ، ماده ضبط کننده ، دستگاه ضبط کننده و یا ماتی به علت ساختار کلی سیستم عوامل ایجاد این امر می باشد.

۱-۲- علم بازیابی تصویر

علم بازیابی تصویر علمی است که به تخمین یا بازیابی اشیاء از یک تصویر تخریب شده توسط نویز و نیز به یافتن عوامل تخریب کننده چه معین و چه آماری می پردازد . با توجه به آنکه نویز تصویر در حالت کلی وابسته به سیگنال است ؛ باید تخمین زننده هایی را که اثر نویز وابسته به سیگنال را در خود در نظر می گیرند ، مورد توجه قرار دارد .

برای هر تصویر اگر بعد سوم در نظر نگیریم به صورت تابعی دو بعدی از X, Y بصورت :
 $U=F(X, Y)$ در نظر گرفته می شود ؛ اگر تصویر به طور دیجیتال Save شده باشد ، که مقدار F در هر نقطه متناسب با روشنایی تصویر عددی بین ۰-۲۵۵ (۸بیتی) برای رنگی و برای سیاه و سفید عددی متناسب با سطح خاکستری است .

پس اگر مقادیر یک تابع تصویر در مختصات سه بعدی بصورت ارتفاع نشان داده شود ، روشنایی تصویر در مختصات سه بعدی بصورت ارتفاع نشان داده خواهد شد ؛ روشنایی تصویر بصورت رویه ای خواهد بود که در این رویه ، نقاطی که روشنایی آنها تغییرات کمی دارد ، صاف و مسطح بوده و در نقاطی که تغییرات روشنایی زیاد است ، تعدادی پیک وجود دارد.

یک تصویر دیجیتال در حقیقت هم در نقاط و هم در روشنایی بصورت نقاطی ناپیوسته است ، بنابراین می توانیم یک تصویر دیجیتال را بصورت یک ماتریس که سطر و ستون آن بیانگر مختصات نقاط در صفحه می باشد ، درآورد و ما هر یک از این نقاط را یک (PIXEL) یا جزء تصویر می نامیم . ابعاد تصویر بهتر است بصورت $(2^M * 2^M)$ باشد ، چون الگوریتم فوریه با این روش تعریف ، بصورت سریع می تواند مورد استفاده باشد .

روشهای تخمین و بازسازی با توجه بر فرض داشتن یک تابع توزیع سیگنال مورد تخریب استوار است و مدل فرض شده بصورت یک فرایند اتفاقی گوسی در نظر گرفته می شود و نویز را بصورت نویز جمع شونده و وابسته به سیگنال در نظر می گیریم: [1]

$$R=CA+KF(A)*n1+n2 \quad (1-1)$$

R: سیگنال اندازه گیری شده حاوی نویز می باشد .

F(A): سیگنال اصلی یا تابعی از آن است .

n1: نویز ضرب شونده با توزیع گوسی .

n2: نویز جمع شونده با توزیع گوسی و مستقل از n1 .

حالا اگر فرض کنیم :

$$C = K = 1 \quad (2-1)$$

$$F(S) = F(A) = A^{0.5}$$

در نتیجه :

$$R=A+A^{0.5} *n1+n2 \quad (3-1)$$

می بینیم که سیگنال و نویز قابل مشاهده اند . چون وابستگی نقاط تصویر در یک همسایگی زیاد است می توان آرگمان سیگنال را بصورت محلی بدست آورد .
برای مشاهدهء روش تخمین باید مراحل زیر انجام شود :

۱- تصویر را نویزی می کنیم :

$$R=A+ A^{0.5}*\sigma1X+\sigma2Y \quad (4-1)$$

$\sigma1,2$: واریانسهای تابع نویز هستند .

۲- اعمال روشهای تخمین .

۳- با اعداد رندم بین صفر و یک با توزیع یکنواخت در متغییر تصادفی X, Y با توزیع نرمال گوسی تولید می کنیم .

در مسایل باینری :

جدول ۱-۱ حالات و نتایج

H_0	H_1	RESULTS	STATES
1	?	H_1	ERROR
?	1	H_0	ERROR
1	?	H_0	TRUE
?	1	H_1	TRUE

۳-۱- معیار بیز [3]

ما در اینجا دو فرض داریم :

اول خروجی منبع است که با یک احتمال زیاد مشخص می شود :

CMN : معرف فرض انتخاب شده است و N معرف آنچه که درست می باشد .

M,N می توانند صفر و یا یک باشند .

P_0, P_1 : دو اطلاعات باینری در قبل و بعد از روشن شدن منبع هستند .

حالا اگر H_0 را انتخاب کنیم و H_1 درست باشد (جمله دوم عبارت) و اگر H_1 را انتخاب کنیم و

H_0 درست باشد (جمله اول عبارت) و همینطور داریم :

$$R = C_{00}P_0P_r + C_{10}P_0P_r + C_{11}P_1P_r + C_{01}P_1P_r \quad (5-1)$$

بدلیل آنکه ما فرض کرده ایم که معیار تصمیم H_1 و یا H_0 را انتخاب می کند فضای مشاهده Z

شامل Z_0 و Z_1 است . وقتی مشاهده در Z_0 باشد و R نقطه ای از فضای مشاهده باشد آنگاه :

$$R = C_{00}P_0 \int_{z_0} P_r / H_0(R / H_0) dR + C_{10}P_0 \int_{z_1} P_r / H_0(R / H_0) dR + C_{11}P_1 \int_{z_1} P_r / H_1(R / H_1) dR + C_{01}P_1 \int_{z_0} P_r / H_1(R / H_1) dR \quad (6-1)$$

فرض کنیم که ارزش تصمیم گیریهای نادرست بیشتر از تصمیم گیریهای درست است آنگاه :

$$\begin{aligned} C_{10} &> C_{00} \\ C_{01} &> C_{11} \end{aligned} \quad (7-1)$$

حال برای بدست آوردن جواب باید ناحیه های Z_0 و Z_1 را طوری انتخاب کنیم که RISK حداقل گردد. برای تصمیم گیری هر نقطه، فضا Z به Z_0 و Z_1 تقسیم می شود:

$$Z = Z_0 \cup Z_1 \quad (8-1)$$

$$\begin{aligned} R = & P_0 C_{00} \int_{z_0} P_r / H_0(R / H_0) dR + P_0 C_{10} \int_{z - z_0} P_r / H_0(R / H_0) dR \\ & + P_1 C_{01} \int_{z_0} P_r / H_1(R / H_1) dR + P_1 C_{11} \int_{z - z_0} P_r / H_1(R / H_1) dR \end{aligned} \quad (9-1)$$

$$\begin{aligned} \int_{z - z_0} P_r / H_0(R / H_0) dR &= 1 - \int_{z_0} P_r / H_0(R / H_0) dR \\ \int_{z - z_0} P_r / H_1(R / H_1) dR &= 1 - \int_{z_0} P_r / H_1(R / H_1) dR \end{aligned} \quad (10-1)$$

$$R = P_0 C_{10} + P_1 C_{11} + \int_{z_0} \{ [P_1(C_{01} - C_{11}) P_r / H_1(R / H_1)] - [P_0(C_{10} - C_{00}) P_r / H_0(R / H_0)] dR \} \quad (11-1)$$

همانطور که مشاهده می شود دو قسمت اول R بدست آمده ثابت است، اما ارزش انتگرال بوسیله نسبت دادن نقاط R به Z_0 کنترل شده است، دو ترم در براکت مثبت هستند، پس تمام مقادیر R که در آن ترم دوم بزرگتر از ترم اول باشد، باید شامل Z_0 گردد و چون مقدار انتگرال منفی است پس تمام مقادیری که انتگرال را مثبت می کند به Z_1 تعلق دارد، در نتیجه اگر:

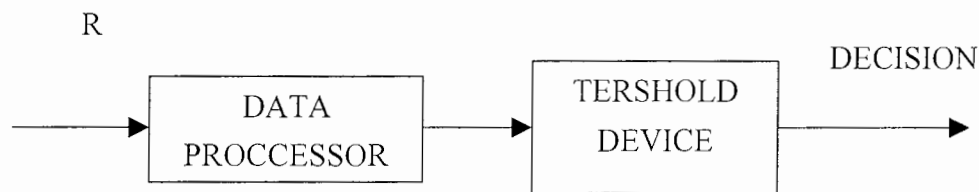
$$P_1(C_{01} - C_{11}) P_r / H_1(R / H_1) \geq P_0(C_{10} - C_{00}) P_r / H_0(R / H_0) \quad (12-1)$$

باشد ، پس R به Z1 نسبت داده می شود و (H1 که درست است) H1 انتخاب می شود در غیر این صورت H0 انتخاب می گردد و آنگاه: [1]

$$\Lambda(R) = \frac{P_r / H_1(R / H_1)}{P_r / H_0(R / H_0)} \quad (13-1)$$

بنابراین معیار بیز نسبت امر احتمالی را نشان می دهد که نامش را نسبت لایک لیهود (LIKELIHOOD RATIO) می نامیم . و به عبارت دیگر آستانه تست :

$$\eta = \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \quad (14-1)$$



شکل ۱-۱- دیاگرام تصمیم گیری با توجه به داده ورودی

۴-۱- تبدیل فوریه [2]

تبدیل فوریه ناپیوسته یک تابع $F(X)$ را تصور کنید که به دسته ناپیوسته :

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2 \times \Delta x), \dots, f(x_0 + (N-1) \times \Delta x)\} \quad (15-1)$$

بوسیله برداشتن N نمونه به فاصله ΔX تبدیل شده است که X بعنوان متغیر ناپیوسته یا پیوسته تعریف می شود ، که ما در اینجا ناپیوسته می گیریم :

$$F(X) = F(X_0 + X * \Delta X) \quad (16-1)$$

که X مقادیر ناپیوسته $0 \dots N-1$ را شامل می شود ، یا به عبارت دیگر :

$$\{f(0), f(1), f(2), \dots, f(N-1)\} \quad (17-1)$$

در نتیجه تبدیل فوریه ناپیوسته :

$$F(u) = (1/N) \times \sum_{x=0}^{N-1} f(x) e^{-j2\pi \times u \times x / N} \quad (18-1)$$

$$u = 0 \dots N-1$$

و معکوس آن بصورت :

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{j2\pi \times u \times x / N} \quad (19-1)$$

$$x = 0 \dots N-1$$

و تبدیل فوریه دو متغیره ناپیوسته :

$$F(U, V) = \left(\frac{1}{MN}\right) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (20-1)$$

$$u = 0 \dots M-1$$

$$v = 0 \dots N-1$$

و معکوسش بصورت :

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(U, V) e^{j2\pi \times (\frac{ux}{M} + \frac{vy}{N})} \quad (21-1)$$

$$x = 0 \dots M - 1$$

$$y = 0 \dots N - 1$$

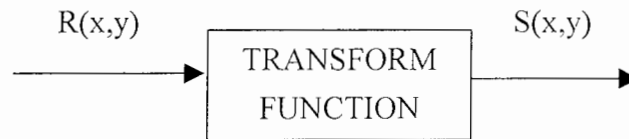
و اگر بردارها مربعی باشند ($M=N$) خواهد شد و تبدیل کمی ساده تر هم می شود .

۱-۵- تخمین تابع تبدیل [1]

ابتداء ورودی r را نویزی می نماییم :

$$r = s + s^{*0.5} * n1 + n2 \quad (22-1)$$

$n1$: نویز وابسته به سیگنال و $n2$ نویز مستقل از سیگنال می باشد ، سپس روی تصویر نویزی یکی از تخمینها را پیاده می کنیم .



شکل ۱-۲- ساختار کلی سیستم

$$S(X, Y) = H(X, Y) * R(X, Y) \quad (23-1)$$

$S(X, Y)$ را نداریم در نتیجه با محاسبه تبدیل فوریه رابطه بالا داریم :

$$H(U, V) = S(U, V) / R(U, V) \quad (24-1)$$

$S(U, V)$: تبدیل فوریه خروجی

$R(U, V)$: تبدیل فوریه ورودی

۱-۵-۱- تبدیل فوریه *CROSS-CORRELATION* :

$$\begin{matrix} S_{sr}(u, v) \\ \{R_{sr}(x, y)\} \end{matrix} \quad (25-1)$$

۱-۵-۲- تبدیل فوریه *AUTO-CORRELATION* :

$$\begin{matrix} S_r(u, v) \\ \{R_r(x, y)\} \text{ or PSD} \end{matrix} \quad (26-1)$$

و درنهایت داریم :

$$H(u, v) = \frac{S_{sr}(u, v)}{S_r(u, v)} \quad (27-1)$$

حال برای تخمین کفایت $R_{sr}(x, y)$ ، $R_r(x, y)$ در یک پنجره مثلثی ضرب گردد ، پس از تخمین تابع تبدیل $H(U, V)$ آنرا در تصویر ورودی ضرب کرده و سپس فوریه معکوس را اعمال کنیم آنگاه خروجی بهبود می یابد .

فصل (۲)

آشنایی بانرم افزار پردازش تصویر

۲- فصل آشنایی با نرم افزار (قسمت پردازش تصویر با Matlab) :

موضوع: آوردن تصویر، کاربر روی تصویر، اعمال نویز و حذف آن.

۲-۱- مقدمه

آوردن تصویر در کامپیوتر :

ابتداء تصویر مورد نظر را scan می کنیم . در نرم افزار Matlab برای اینکه روی یک تصویر (Image) کار کنیم .اول باید آن را بصورت فایل *.Tif* در آوریم. این کار با Save as تصویر در Acd see (یا نرم افزارهای مشابه گرافیکی) میسر خواهد شد.

۲-۲- شروع عملیات پردازش تصویر

حالا شروع به انجام عملیات پردازش تصویر در ابعاد مختلف می نماییم از آن جمله:

۲-۲-۱- خواندن تصویر (Read Image) با دستور Imread .

۲-۲-۲- کشیدن تصویر اصلی (Original Image) بادستور Imshow .

۲-۲-۳- تغییر دادن اندازه تصویر (Resize The Image) بادستور Imresize .

۲-۲-۴- انتخاب یک زیر ناحیه از شکل (Choose Subregion of Each Image) با دستور

. Imcrop

۲-۲-۵- کاهش لرزش لبه ها (Reduce Edge Ringing) با دستور Edgetaper .

۲-۲-۶- شبیه سازی یک تصویر بلوری و نویزدار (Simulate Bluer & Noisy Image) .

۲-۲-۷- بازسازی تصویر نویزی و بلوری (Restore The Blurred & Noisy Image) .

۲-۲-۸- مشاهده اثر تعداد تکرار بر کارکرد بازسازی تصویر .

۲-۲-۹- چرخاندن تصویر با یک زاویه مشخص بر حسب درجه (Rotate The Image) با

دستور (Imrotate) .

حال به توضیح خلاصه ای از هربخش به همراه نرم افزارش می پردازیم :

۲-۲-۱- همان طور که از نامش پیداست کارخواندن فایل تصویری Tif* را انجام می دهد.

```
I = imread('cameraman.tif')
```

۲-۲-۲- در این قسمت شکل اصلی و خوانده شده در قسمت قبل را نمایش می دهیم ؛ البته به عنوان شکل اصلی :

```
figure;
```

```
imshow(I);
```

```
title('Original Image');
```

۲-۲-۳- تغییر دادن اندازه تصویر که به (Scaling) تصویر هم مشهور است به این صورت است که:

```
k=imresize(I,7);
```

```
figure;
```

```
imshow(K);
```

```
title('Resized Image');
```

۲-۲-۴- انتخاب زیر ناحیه ای از شکل که با دستور برش یا Imcrop انجام می شود به دو صورت:

```
Rectcam=[93 13 81 69];
```

```
Subcam=imcrop(I,Rectcam);
```

OR

```
[Subcam,Rectcam]= Imcrop( I)
```

Subcam: معرف زیر ناحیه ای از شکل اصلی است .

Rectcam: محدودهء برش را مشخص می کند .

I: شکل اصلی است .

۲-۲-۵- کاهش لرزش (Ringing) لبه های تصویر : ممکن است تصویری که در اختیار ما است دارای کمی لرزش در لبه های اجزاء تصویر باشد که با دستور `Egdetaper` آن را حذف کرده و یا کاهش می دهیم . (یعنی حذف دو یا چند تا دیده شدن مرزها ، در اجزاء تصویر) به این صورت که :

```
Edged=edgetaper(BlurredNoisy,PSF);
```

```
Shap2=deconvlucy(Edged,PSF,5);
```

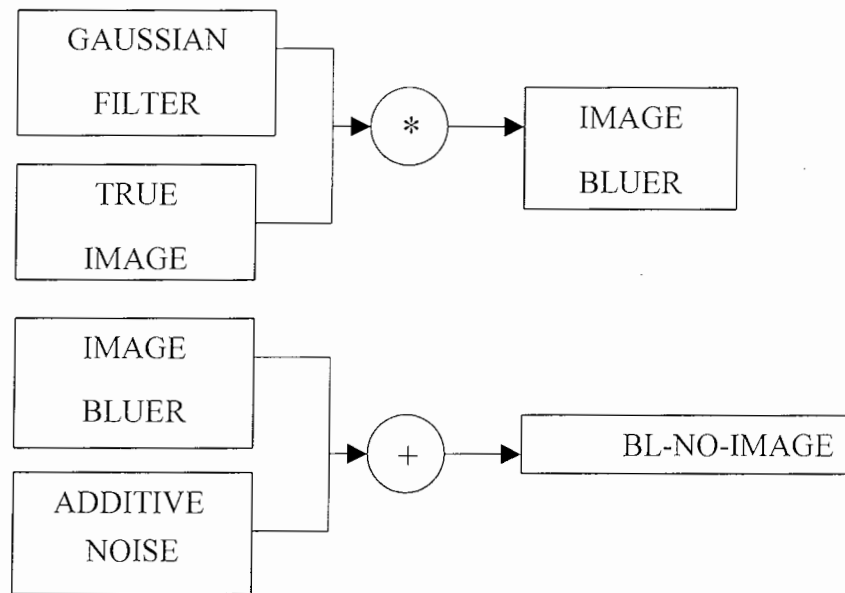
```
figure;
```

```
imshow(Shap2);
```

```
title('Edgetaper effect');
```

۲-۲-۶- شبیه سازی یک تصویر بلوری و نویزدار: تصویرهایی که از دنیای واقعی دریافت می شوند ، به علت حرکت دوربین یا عدم دقت (Focus & Resolution) مناسب دارای درصدی بلوری شدن هستند و همچنین این تصویرها به علت وجود معایب یا کاستیهایی از حالت ایده ال برای دستگاه فیلم برداری ، دارای نویز هستند ، که ما این حالت بلوری شدن (حرکت داشتن تصویر) و همچنین نویز موجود در دستگاههای مختلف تصویر برداری یا محیطهای مختلف را با یک اعوجاج رندم و تصادفی مدل می کنیم. این شبیه سازی بوسیله یک عمل کانولوشن بین تصویر اصلی و فیلتر یا تابع گوسی صورت می گیرد ، که با دستور `Imfilter` انجام می شود.

تابع گوسی بوسیله یک تابع گسترش نقطه (Point-spread-Function) مدل می شود . عملیات به شکل زیر است که ابتداء شکل اصلی را بلوری کرده و نویز خود را جمع پذیر در نظر گرفته و شکل را بعد از بلوری شدن نویزدار می کنیم :



شکل ۱-۲- ساختار الگوریتم بلوری و نویزی نمودن تصویر

بکمک دستورات زیر :

```

PSF = fspecial('gaussian',7,10);
Blurred = imfilter(I,PSF,'symmetric','conv');
figure;
imshow(Blurred);
title('Blurred Image');

V = .001;
BlurredNoisy = imnoise(Blurred,'gaussian',0,V);
figure;
imshow(BlurredNoisy);
title('Blurred & Noisy' Image);

```

۷-۲-۲- بازسازی و بازیابی تصویری که نویزی و بلوری شده است راههای مختلفی دارد از آن جمله :

۱-۷-۲-۲- دی کانولوشن با همان تابعی که شکل بلوری و نویزی شده:

در این روش تعداد تکرار مهم است ، اگر تعداد تکرار را بالا ببریم فیلتر ما در حوزه فرکانس Sharp شده و فرکانسهای بالا که در آن محل است را از بین می برد ، در نتیجه شکل بهتری داریم :

```
Shapmainway1 = deconvlucy(BlurredNoisy,PSF,15);
```

```
Shap1 = im2uint8(Shapmainway1);
```

```
figure;
```

```
imshow(Shap1);
```

```
title('Restored Image, NUMIT =15');
```

۲-۷-۲-۲- بکار بردن منبع نویز به عنوان ورودی سوم :

در این روش ما یک منبع نویز ایجاد می کنیم و به این ترتیب می نویسیم :

```
NP=V * prod(size(I));
```

```
[reg1 LAGRA]=deconvreg(BlurredNoisy,PSF,NP);
```

```
figure;
```

```
imshow(reg1);
```

```
title('RESTORED WITH NP');
```

۳-۷-۲-۲- بکار بردن ضرب کننده لاگرانژ (ضریب لاگرانژ) :

در ادامه روش دوم ، LAGRA را برای این بدست آوردیم که بشود از آن پارامتر در این روش استفاده کنیم ، یعنی ضریب وابستگی ضرب کننده لاگرانژ را داشته باشیم ، همچنین فرض بر این است که جواب ایده آل نیز موجود باشد :

LAGRA : شکل اصلی را می دهد .

و $b > 1$ LAGRA*b : با این تغییر باعث ضعیفتر شدن قید در نتیجه نرم شدن (تاری) شکل.

و $b < 1$ LAGRA*b : با این تغییر باعث قویتر شدن قید در نتیجه سخت شدن (بهبود) شکل.

```
Edged=edgetaper(BlurredNoisy,PSF);
reg2=deconvreg(Edged,PSF,[],LAGRA/90);
figure;
imshow(reg2);
title('RESTORED WITH LAGRANGE MULTIPLIER');
```

۲-۲-۴- بکاربردن آرایهٔ وزنی به این شکل که :

ابتداءً آرایهٔ وزنی را طوری تعریف می‌کنیم که لبه‌ها را از بین ببرد و عمل Deconvolution تصویر بلوری و نویزی حاصله را با آرایهٔ وزنی انجام می‌دهیم ، با انجام مراحل زیر شکل اصلی بدون لرزش و نویز و بلور بدست می‌آید . البته ، اگر تعداد تکرار این عمل را زیاد کنیم ، (مثلاً ۳۰ بار) :

```
Weight = edge(I,'sobel',.3);
se = strel('disk',2);
Weight = 1-double(imdilate(Weight,se));
```

حال اگر آرایهٔ وزنی را طوری تعریف کنیم که در نزدیکی لبه‌های اجزاء تصویر (چون پیکسل‌های نامناسب داریم) دارای مقدار صفر شود؛ آنگاه :

```
Weight([1:3 end-[0:2]],:) = 0;
Weight(:, [1:3 end-[0:2]]) = 0;
figure;
imshow(Weight);
```

```
title('Weight array');
```

در نتیجه تصویر نهایی که از دی کانولوشن کردن این آرایه با تعداد تکرار زیاد (۳۰ بار) با تصویر بلوری و نویزی حاصل می شود؛ تقریباً بدون بلور و نویز خواهد بود، یعنی معادل شکل اصلی:

```
[J P] = deconvblind(Blurred,INITPSF,30,[],Weight);
```

```
figure;
```

```
imshow(J);
```

```
title('Restored Image WITH WEIGHT ARRAY');
```

۲-۲-۷-۵- فیلتر معکوس به همراه کنترلر دامنه نویز، تولید یک نسبت سیگنال به نویز (NSR) :

این روش هم مثل روش اول بوده و تنها تفاوتش در این است که دامنه نویز کنترل می

- شود.

$$R = \frac{|N^2|}{|S^2|}$$

```
noise = 0.15 * randn(size(I));
```

```
NSR = sum(noise(:).^2)/sum(im2double(I(:).^2));
```

```
Shap4= deconvwnr(BlurredNoisy,PSF,NSR);
```

```
figure;
```

```
imshow(Shap4);
```

```
title('Restored with NSR');
```

۲-۲-۸- مشاهده اثر تعداد تکرار بر کارکرد بازیاب تصویر: همانطور که گفته شد در بخش بازسازی روش اول اثر زیاد شدن تعداد تکرار (*Number Of Iterate*) این است که باعث بهبود پاسخ فرکانسی (حذف بهتر فرکانسهای بالا و نویز) و ایده‌آل‌تر شدن پاسخ فرکانسی و تندتر شدن آن است و محصول آن، شکل مناسب‌تر می‌باشد.

۲-۲-۹- چرخاندن تصویر با یک زاویه خاص بر حسب درجه: توسط دستور زیر:

```
L=imrotate(K,30);
```

```
figure;
```

```
imshow(L);
```

```
title('Rotated Image');
```

(چرخاندن تصویر K به اندازه ۳۰ درجه و ذخیره‌سازی شکل در L).

۲-۳- برنامه جامع و کاربردی پردازش تصویر

در قسمت ضمیمه (۱) ما با بکاربردن تمامی این دستورات و زیر روالها برنامه‌ای جامع در مورد عملیتهای پردازش تصویر (Image processing) و نیز اثرات آنها بر تصویر دریا فتی را آرایه خواهیم کرد و با توجه به نتایج حاصله بهترین روش بازیابی تصویر را از نظر کیفیت معرفی می‌کنیم.

۲-۴- نتیجه:

از مشاهده کل این روشها به این نتیجه رسیدیم که با توجه به کیفیت تصویر خروجی رتبه بندی

روشها به این صورت می‌باشد: ۱- آرایه وزنی

۲- کانولوشن با تابع گوسی

۳- لاگرانژ

فصل (۳)

تبدیل‌ها و فشرده‌سازی تصاویر

۱-۳- فشرده سازی تصویر

تبدیل هارتلی HT (که برای رشته هایی با طول زیاد کار برد دارد .)

تبدیل فوریه FT و مقایسه با تبدیل هارتلی .

تبدیل کسینوس DCT (که با نرم افزار مطلب ، با هر تعداد ضریب تبدیل، عمل فشرده سازی را می-

توان انجام داد .)

تبدیل Cas-Cas .

۲-۳- چکیده فصل

نتایج بدست آمده نشان می دهد که تبدیل هارتلی از لحاظ فشرده سازی بهتر از تبدیل فوریه و تبدیل کسینوس بهتر از دو تبدیل فوریه و هارتلی عمل می کند . تبدیل Cas-Cas از نظر محاسباتی ساده تر از تبدیل هارتلی است و بهتر است در فشرده سازی از آن به جای تبدیل هارتلی استفاده شود .

تبدیل هارتلی یک تبدیل حقیقی است که به علت خواص خوبش مورد بحث قرار گرفته است . چون اکثر تبدیلهای حجیم هستند و عملاً غیر قابل استفاده می باشند ، پس نیاز به الگوریتم سریع محاسباتی داریم ، که الگوریتمهای سریعی چه در حوزه فرکانس و چه در حوزه زمان برای تبدیل هارتلی ارائه شده است ، الگوریتمهایی که هم از سرعت Radix-4 بهره می گیرند و هم خواص Radix-2 را دارند و پیاده سازی راحتی دارند . همان طور که گفته شد الگوریتمها برای رشته هایی با طول زیاد کاربرد دارند ، اگر الگوریتمها را پیچیده تر انتخاب کنیم سرعت بیشتر هم می شود و تعداد ضربها به کمتر از نصف کاهش می یابد .

مشکلات محاسباتی و همچنین دشواری پاره ای از مسایل تحلیلی دانشمندان را بر آن داشت تا در صورت امکان مسایل خود را در فضای دیگری که در تناظر یک به یک با فضای اصلی باشد و در عوض سهولت راه حل را بدنبال داشته باشد ، مورد بررسی قرار دهند . در این راستا تبدیلات مختلفی از

جمله تبدیل فوریه، لاپلاس، Z ، هارتلی و معرفی شد. یکی از خصوصیات مهم این تبدیلهای جدا سازی اطلاعات موجود در مجموعه داده ها می باشد.

تبدیلات مرتبط به فوریه در تکنولوژی، صنعت و علمی - مهندسی کاربرد زیادی پیدا کرده اند. هر تابع دلخواهی را می شود بصورت توابع مثلثاتی در آورد، ولی چون محاسبات مختلط می شود حافظه و زمان زیادی را می طلبد. در صورتی که خود تابع حقیقی باشد، مثلاً "برای صوت، گوش ما به فاز حساس نیست و اطلاعات فاز قابل صرف نظر می باشد.

۳-۳- هارتلی [7]

در سال ۱۹۴۸ هارتلی به معرفی تبدیل پرداخت که هسته آن را $(\cos x + \sin x)$ تشکیل می-داد.

۳-۳-۱- مزایای این تبدیل :

۱- حقیقی بودن

۲- مشابه بودن هسته تبدیل به تبدیل فوریه مشهور

۳- معکوس ساده اش بود.

از آنجا که هسته تبدیل شباهت زیادی به هسته تبدیل فوریه داشت و با توجه به اینکه تبدیل از قدمت بیشتری برخوردار بود توجهی به این تبدیل نشد و به فراموشی سپرده شد تا اینکه در دو دهه اخیر برسیول (BRACEWELL) آن را دوباره زنده کرد و کتابی به عنوان تبدیل هارتلی چاپ نمود. یکی دیگر از عوامل بی توجهی به این تبدیل، وجود تبدیل کسینوس بود که نسبت به تبدیل هارتلی از سرعت بیشتری برخوردار بود، ولی چون جمله سینوس نداشت تفاوتی با تبدیل هارتلی داشت.

برای کاهش محاسبات دانشمندان در سال ۱۹۶۰ اولین الگوریتم سریع محاسباتی را برای تبدیل فوریه گسسته توسط کولی - تاکی (COOLY_TUKEY) معرفی کردند ، که تعداد محاسبات $(+ \text{ و } *)$ از N^2 به $N \log N$ کاهش یافت (N تعداد داده های ورودی تبدیل می باشد) ، بعد از آن برای تمام تبدیلهای الگوریتمهای سریع و متفاوتی با روشهای مختلف ارائه شد .

اولین آن در سال ۱۹۸۴ توسط برسیول درست شد که این روش ، روش کولی- تاکی برای (FFT) بود و بعدها هو (HOU) ، مالوار (MALVAR) راههای دیگری را ارائه کردند .

امروزه استفاده از اطلاعات تصویری بسیار متداول شده است و پردازش تصویر (IMAGE PROCESSING) یکی از شاخه های پردازش سیگنال معرفی می شود .

۳-۴- ملزومات سیستم های تصویری [9]

ملزومات سیستم های تصویری انتقال و ذخیره سازی تصویر می باشد . حجم بالای اطلاعات تصویری محدودیت جدی در هر دو مورد ایجاد کرده است . محدودیت حافظه ، پهنای باند کانال انتقال و زمان و انرژی اهمیت کاهش حجم اطلاعات یا به عبارت دیگر فشرده سازی تصویر را نمایان کرده است . در فشرده سازی داده ها سعی می شود که داده های زاید دور ریخته شوند و حتی بعضی از داده های اصلی که حذفشان به کیفیت تصویر لطمه زیادی وارد نمی کند نیز دور ریخته می شوند . برای این کار از روشهای مختلفی استفاده می شود یکی از قدرتمند ترین روشها روش تبدیل خطی است که در این روش حجم زیادی از اطلاعات در تعداد کمتری از ضرایب تبدیل ذخیره می شود و بقیه ضرایب بی اهمیت را می توان حذف نمود .

در مورد فشرده سازی تصویر با استفاده از تبدیل فوریه و کسینوس تحقیقات زیادی شده است از جمله MPEG و JPEG که بر مبنای DCT می باشند ، بی نظیرند .

ولی در مورد فشرده سازی تصویر با تبدیل هارتلی تحقیقات چندانی صورت نگرفته است ، تنها چند مورد فشرده سازی تصویر پزشکی داریم .

جان ویلاسز مقاله ای در سال ۱۹۹۳ منتشر کرد که نشان داد اگر مدل مناسبی برای تصویر پزشکی انتخاب شود ، تبدیل هارتلی برای فشرده سازی مناسب تر از تبدیل کسینوس خواهد بود . چون ساختار فشرده سازی تصاویر پزشکی متفاوت از تصاویر معمولی است پس باید عملکرد هارتلی را در مورد تصاویر معمولی بررسی کنیم .

۳-۵- معرفی تبدیل هارتلی [7]

برای کارهای مختلف وحل مسایل علمی- مهندسی در نهایت ما با فیلتر دیجیتال ، کانولوشن ، همبستگی (CORRELATION) ، پردازش تصویر ، معادلات با مشتقات جزئی ، سیگنالها و توابع حقیقی مواجه هستیم .

اگر از تبدیل مختلط استفاده کنیم حجم کار زیاد شده و می بایستی حافظه زیادی را بکار بریم و متعاقباً زمان زیادی را تلف می کند ، تازه اگر سیگنال چند بعدی هم باشد مشکل جدی تر خواهد بود . در نتیجه دانشمندان به فکر استفاده از توابع حقیقی افتادند ، که لازمه آن وجود یا پیدا کردن تبدیلهای حقیقی است .

۳-۵-۱- تاریخچه هارتلی

در سال ۱۹۴۸ هارتلی تبدیل حقیقی را پیدا کرد و مقاله ای در مجله مهندسی :

PROCEEDING OF THE INSTITUTE OF RADIO ENGINEERS چاپ نمود و در آن هسته

تبدیلش را بصورت :

$$\text{Cas}(x) = \text{Cos } x + \text{Sin } x \quad (1-3)$$

معرفی کرد .

هارتلی یکسان بودن هسته این تبدیل و عکس آن ، که در مرحله آنالیز و سنتز بکار می رود را بعنوان شاخص معرفی کرد . هسته این تبدیل شبیه به FT بوده و هسته تبدیل فوریه بصورت :

$$\text{ex}(jx) = \text{Cos } x + j \text{Sin } x \quad (2-3)$$

برسیول همانطور که گفته شد الگوریتمهای سریعی مثل FFT به نام FHT :

(FAST HARTLEY TRANSFORM) ، ایجاد کرد . چون تبدیل هارتلی حقیقی است و کامپیوتر برای ذخیره سازی اطلاعات مختلط دو برابر اطلاعات حقیقی حافظه نیاز دارد و برای محاسبات مختلط زمان بیشتری نیاز دارد ، پس منطقی است که در مواردی که بحث از سیگنال حقیقی می باشد تبدیل فوریه با راندمان پایین عمل کند . پس تبدیلات حقیقی مشابهی از جمله هارتلی مورد توجه واقع می-شود تا زمان و حجم محاسبات و همچنین حافظه مورد نیاز کاهش یابد .

همانطور که گفته شد چون تبدیل DCT (تبدیل کسینوس گسسته) ، دارای هسته $\cos X$ است و از آنجائیکه این تبدیل هم تبدیل حقیقی است ، نسبت به فوریه و هارتلی از سرعت بالاتری برخوردار است و چون در اکثر موارد زمان حقیقی کاربرد دارد ، این روش بسیار بکار می رود . اما چون جمله $\sin X$ را ندارد ممکن است نسبت به هارتلی کاستیهایی داشته باشد .

تبدیل هارتلی توابع حقیقی بیشتر از تبدیل هارتلی توابع مختلط مورد توجه هستند . که تابعی از متغیر حقیقی U هستند .

۳-۶- توابع مختلط [5]

تبدیل فوریه:

$$F(u) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi ut} dt \quad (3-3)$$

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt < \infty \quad (4-3) \quad \text{به شرط محدود بودن انرژی سیگنال :}$$

یا شروط دریکله :

$$\int_{-\infty}^{+\infty} |F(X)| dx < \infty \quad 1-$$

۲- $F(X)$ دارای تعداد محدودی MAX,MIN در هر بازه محدود باشد .

۳- $F(X)$ تعداد محدودی ناپیوستگی در هر بازه محدود داشته باشد و هر ناپیوستگی هم باید محدود باشد .

پس هر سیگنال که :

۱- قدر مطلق آن انتگرال پذیر .

۲- پیوسته .

۳- تعداد محدودی ناپیوستگی داشته باشد .

آنگاه: **تبدیل فوریه** دارد . این سه شرط دقیقاً برای تبدیل هارتلی هم متصور است و معکوس تبدیل فوریه:

$$f(t) = \int_{-\infty}^{+\infty} F(u) e^{i2\pi ut} du \quad (5-3)$$

می باشد .

اگر توابع مهندسی باشند اکثراً $f(t)$ محدود است ولی انرژی نامحدود دارد (یعنی تابع پریودیک است.) و شرط تبدیل پذیری را ندارد برای این کار باید روی یک دوره کار کرد .

۳-۷- **تبدیل هارتلی $H(u)$: [7]**

تبدیل هارتلی سیگنال $f(t)$:

$$H(u) = \int_{-\infty}^{+\infty} f(t) [\cos(2\pi ut) + \sin(2\pi ut)] dt \quad (6-3)$$

وهستهء تبدیل :

$$\text{Cas}(t) = \cos t + \sin t = \frac{(1-i)}{2} e^{it} + \frac{(1+i)}{2} e^{-it} \quad (7-3)$$

$$e^{(it)} = \frac{(1+i)}{2} \cos t + \frac{(1-i)}{2} \cos(-t)$$

$$\int_{-\infty}^{+\infty} \text{Cas}(2\pi ut) \times \text{Cas}(2\pi u\tau) du = \delta(t - \tau) \quad (۸-۳)$$

در نتیجه ، هارتلی و فوریه :

$$\begin{cases} H(u) = \frac{(1+i)}{2} F(u) + \frac{(1-i)}{2} F(-u) \\ F(u) = \frac{(1-i)}{2} H(u) + \frac{(1+i)}{2} H(-u) \end{cases} \quad (۹-۳)$$

این روابط متقارنند و اطلاعات یکسان دارند ، در بحث توابع مختلط تبدیل هارتلی هیچ مزیتی بر تبدیل فوریه ندارد و خواص تبدیل فوریه و هارتلی مشابه اند و تنها روابط * و * آن پیچیدگی بیشتری نسبت به تبدیل فوریه دارد و برای تحلیل سیستمهای خطی این یک عیب برای تبدیل هارتلی بشمارمی آید .

۳-۷-۱- ضرب در حوزة زمان : $V1(t).V2(t)$

فوریه : $F1(f)*F2(f)$

هارتلی :

$$\frac{1}{2}[H_1(f)*H_2(f)+H_1(-f)*H_2(f)+H_1(f)*H_2(-f)-H_1(-f)*H_2(-f)] \quad (۱۰-۳)$$

۳-۷-۲- کانولوشن در حوزة زمان : $V1(t)*V2(t)$

فوریه : $F2(f). F1(f)$

هارتلی :

$$\frac{1}{2}[H_1(f)H_2(f)-H_1(-f)H_2(-f)+H_1(f)H_2(-f)+H_1(-f)H_2(f)] \quad (11-3)$$

ولی اگر یکی از V_1, V_2 ها متقارن باشد روابط ساده تر می شود .

۳-۷-۳- روابط حوزه زمان وفرکانس فوریه وهارتلی [7]

و رابطه ری لی (Ray Leigh) :

$$H(0) = F(0) = \int_{-\infty}^{+\infty} V(t)dt \text{ (Infinite integral)} \quad (12-3)$$

$$\int_{-\infty}^{+\infty} H(f)H^*(f)df = \int_{-\infty}^{+\infty} F(f)F^*(f)df = \int_{-\infty}^{+\infty} |V(t)|^2 dt$$

همانطور که اشاره شد مزیت هارتلی وقتی ظاهر می شود که تابع $f(t)$ حقیقی باشد در نتیجه $H(u)$ حقیقی شده و آنگاه تبدیل ملموس تر و میل به ایجاد الگوریتمهای سریع تر آن برای جایگزینی اش بجای تبدیل فوریه بیشتر می شود .

تبدیل فوریه یک تابع حقیقی بطور کلی مختلط می باشد ، پس $F(u)$ شامل اطلاعات زایدی خواهد بود در واقع $F(u)$ بوسیله مقادیر آن بر روی نیم خط $u \geq 0$ و یا $u \leq 0$ کاملاً مشخص است .

با جایگزینی در روابط :

$$H(u) = \frac{(1+i)}{2} F(u) + \frac{(1-i)}{2} F(-u) \quad (13-3)$$

$$, F(-u) = F^*(u)$$

در نتیجه :

$$H(u) = R\{F(u)\} - I\{F(u)\} \quad (14-3)$$

ولی هیچ ارتباط تقارنی بین $H(u)$, $H(-u)$ وجود ندارد و همه مقادیر موجود روی محور حقیقی برای مشخص کردن نیاز است ، بنابراین رابطه ساده تر نشد . اما اگر تبدیل هارتلی را برای صفحه

اعداد مختلط تعریف کنیم (نگاشت بدهیم)، u به $w=u+iv$ تبدیل می شود (چون برای ما مهم است، که سیگنال حقیقی یا طیفش را روی صفحه مختلط نمایش دهیم).

۳-۸- توابع مختلط [7]

فوریه :

$$F(w) = \int_{-\infty}^{+\infty} f(t)e^{(-i2\pi wt)} dt \quad (15-3)$$

هارتلی :

$$H(w) = \int_{-\infty}^{+\infty} f(t)[\cos(2\pi wt) + \sin(2\pi wt)] dt \quad (16-3)$$

در نتیجه :

$$H(w) = \frac{(1+i)}{2} F(w) + \frac{(1-i)}{2} F(-w) \quad (17-3)$$

اگر $f(t)$ حقیقی باشد :

$$F(w^*) = F^*(-w)$$

یعنی $F(w)$ در u تقارن هرمیتی دارد (یعنی بر روی محور موهومی خاصیت انعکاسی دارد).

$$H(w) = \frac{(1+i)}{2} F(w) + \frac{(1-i)}{2} F^*(w^*) \quad (18-3)$$

همچنین دیده می شود که $H(w^*) = H^*(w)$ یعنی $H(w)$ در v تقارن هرمیتی دارد، یعنی خاصیت انعکاسی بر روی محور حقیقی و چون خاصیت انعکاسی دارد، آنگاه خط راست L یا (محور موهومی و یا محور حقیقی) هم به همین صورت نگاشته می شود. پس نگاشت ما نوعی تبدیل خطی می باشد. آنگاه می توان برای $f(x)$ حقیقی قسمتهای حقیقی و موهومی $H(w)$ را بصورت زیر جدا کرد:

$$H(w) = H(u, iv) = \int_{-\infty}^{+\infty} f(x) \cosh(2\pi vx) [\cos(2\pi ux) + \sin(2\pi ux)] dx +$$

$$i \int_{-\infty}^{+\infty} f(x) \sinh(2\pi vx) [\cos(2\pi ux) - \sin(2\pi ux)] dx \quad (19-3)$$

تبدیل هارتلی هم مثل تبدیل لاپلاس یا فوریه توانایی دو یا چند بعدی شدن را دارد، مثلاً تبدیل دو بعدی هارتلی:

$$H(u_1, u_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \text{Cas}[2\pi(u_1 x + u_2 y)] dx dy \quad (20-3)$$

است و معکوسش:

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} H(u_1, u_2) \text{Cas}[2\pi(u_1 x + u_2 y)] dx dy \quad (21-3)$$

در نهایت مثل تک بعدی اگر $f(x, y)$ حقیقی باشد:

$$H(u_1, u_2) = R\{F(u_1, u_2)\} - I\{F(u_1, u_2)\} \quad (22-3)$$

۹-۳- تبدیل (Cas-Cas) T [10]

تبدیل هارتلی دو بعدی مثل تبدیل فوریه دو بعدی قابل جدا شدن به تبدیلات تک بعدی نیست.

ولی می دانیم که هسته تبدیل فوریه دو بعدی بصورت زیر قابل جدا سازی است:

$$\exp[-i2\pi(u_1x + u_2y)] = \exp[-i2\pi u_1x] \exp[-i2\pi u_2y] \quad (23-3)$$

در حالیکه برای هسته هارتلی چنین چیزی ممکن نیست :

$$\text{Cas}[2\pi(u_1x + u_2y)] \neq \text{Cas}(2\pi u_1x) \text{Cas}(2\pi u_2y) \quad (24-3)$$

از آنجایی که خاصیت جدا سازی شرایط خاصی را برای محاسبات عددی تبدیلات گسسته ایجاد می کند در سال ۱۹۸۶ برسیول و همکارانش به معرفی تبدیلی تحت عنوان Cas-Cas پرداختند و نشان دادند که توسط چند جمع و تفریق ساده می توان تبدیل هارتلی را از آن محاسبه نمود .

تبدیل T or Cas-Cas :

$$T(u_1, u_2) = \int_{-\infty-\infty}^{+\infty+\infty} f(x, y) \text{Cas}(2\pi u_1x) \text{Cas}(2\pi u_2y) dx dy \quad (25-3)$$

عکس تبدیل T مشابه خود تبدیل T می باشد و تبدیلی با استفاده از تقارنهای و داشتن تبدیل

T بصورت زیر محاسبه می شود :

$$H(u_1, u_2) = \frac{1}{2} [T(u_1, u_2) + T(u_1, -u_2) + T(-u_1, u_2) - T(-u_1, -u_2)] \quad (26-3)$$

ولی باز هم این تبدیل در کانولوشن به علت داشتن محاسبات زیاد کارایی کمتری پیدا می کند .

۳-۱۰- تبدیل هارتلی گسسته DHT: [7]

چون در محاسبات با کامپیوتر دیجیتال نیاز به گسسته سازی داریم پس DHT را بحث می کنیم :

فوریه گسسته :

$$F(u) = (1/N) \times \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi \times u \times x}{N}} \quad (27-3)$$

$$u = 0 \dots N-1$$

و معکوس آن :

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi \times u \times x}{N}} \quad (28-3)$$

$$x = 0 \dots N - 1$$

که در این دو فرمول N تعداد نمونه ها و یا داده ها می باشد.

و هارتلی و معکوس آن بصورت :

$$H(U) = \frac{1}{N} \sum_{n=0}^{N-1} f(x) \text{Cas}(2\pi un / N) \quad (29-3)$$

$$f(n) = \sum_{u=0}^{N-1} H(u) \text{Cas}(2\pi un / N)$$

مثال : تبدیل هارتلی تابع $V(t)$ (تابع دروازه) :

$$V(t) = \begin{cases} 1 \rightarrow -0.5 \leq t \leq 0.5 \\ 0 \rightarrow elsewhere \end{cases} \quad (30-3)$$

$$H(f) = \int_{-\infty}^{+\infty} V(t) \text{Cas}(2\pi ft) dt = \frac{1}{2\pi f} [\sin(2\pi f) - \cos(2\pi f) + 1] \quad (31-3)$$

و فوریه آن :

$$F(f) = \frac{\sin(2\pi f)}{(2\pi f)} - j \frac{(1 - \cos(2\pi f))}{(2\pi f)} \quad (32-3)$$

۱۱-۳ - تبدیل هارتلی سریع FHT: [7]

باتوجه به وسعت کاربرد برای سیگنالهای حقیقی به فکر سرعت بخشیدن به محاسبات هارتلی می افتیم . اولین الگوریتم سریع DFT ، کولی- تاکی است که در سال ۱۹۶۰ بصورت radix-2 (رشته ورودی توانی از دو باشد) نوشته شد . در این روش از شیوه نصف کردن استفاده می شود (در کل اگر تعداد ورودیهای N یک تبدیل N باشد و بتوان $N=B^m$ نوشت آنگاه اطلاعات از نوع radix-B است) .

مثلاً برای DFT (N تعداد داده باشد) آنگاه تعداد عملیات (+ و *) از N^2 به $N \log N$ تقلیل می یابد، یعنی برای اجرای این الگوریتمهای سریع اطلاعات را بصورت خاصی دسته بندی می کنند اگر دسته بندی در حوزه زمان باشد آنرا Decimation In Time (DIT) و اگر در حوزه فرکانس باشد، Decimation In Frequency (DIF) می نامند.

برای هارتلی اولین الگوریتم سریع (FHT) در سال ۱۹۸۶ بوسیله برسیول ایجاد شد، این الگوریتم از نوع DIT و Radix-2 (مثل روش کولی- تاکی برای FFT) است، یکی دیگر از روشها روش هو (Hou) بود، که ماتریس تبدیل هارتلی را به ماتریسهایی با ابعاد کوچکتر تقسیم کرد و چون محاسبات برای ماتریسهای کوچک آسان تر است، در نتیجه تعداد محاسبات کاهش می یابد و می توان به سرعتهای بالاتری دست یافت.

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \end{bmatrix} = T'(4) \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \quad (3-33)$$

f: تابع ورودی .

H: تبدیل هارتلی .

T'(4): ماتریس تبدیل است .

حال به روش (هو) داریم :

$$T'(4) = \begin{bmatrix} T(2) & T(2) \\ T(2) & -T(2) \end{bmatrix} \quad (3-34)$$

که ماتریس 4×4 به 4 ماتریس 2×2 مشابه هم تبدیل شد این روند اساس کار الگوریتم "HOU" می باشد، یکی دیگر از الگوریتمهای سریع محاسباتی FHT الگوریتم مالوار است که در آن تبدیل

هارتلی N نقطه ای به هارتلی $N/2$ نقطه ای (دوتبدیل کسینوس) تبدیل می شود و همین روش برای تبدیل سریع کسینوسی قابل استفاده است.

اما ناگشا (Nagesha) در سال ۱۹۸۹ اثبات کرد این روش برای تبدیل سریع کسینوسی مناسب نیست و تنها برای هارتلی مناسب است و بعدها روشهایی بر مبنای $Radix2, Radix3, Radix3/9$ ایجاد شد، که سرعت را بالاتر هم برد.

برسیول در کتاب Two-Dimensional Imaging بیان داشت که با توجه به افزایش سرعت کامپیوترها بصورت مضربی از دو تلاش برای افزایش سرعت زیاد مهم نیست، بلکه باید دیگر ب فکر ساده سازی الگوریتم و کم کردن مقدار حافظه مورد نیاز باشیم.

۳-۱۲- فشرده سازی تصویر [8]

مقدمه :

با توجه به نیازهایی که در علوم به فشرده سازی پیدا شد (محدودتهای زمانی- پهنای باند کانال محدود - نیاز به سرعت بالا در انتقال و کار با تصویر و انتقال اطلاعات تصویری) فشرده سازی اهمیت زیادی پیدا کرد.

مثال :

فرض کنیم که یک تصویر با ابعاد $256 * 256$ و حد تفکیک (resolution) 256 سطح خاکستری ۸ بیتی به حافظه ای با حجم $(256 * 256 * 8)$ معادل 512 کیلو بایت نیاز دارد، همچنین اگر ب فرض 30 عدد از این تصاویر را بخواهیم ارسال کنیم به نرخ ارسال 15Mbit/S نیاز خواهیم داشت، که ارسال آن با خط تلفن غیر ممکن است. کاربرد فشرده سازی در ارسال و ذخیره سازی اطلاعات است، در بخش تلویزیونی- دریافت تصاویر از ماهواره- ارتباط نظامی از طریق رادار- سونار- کنفرانس از راه دور - تلفن تصویر مورد استفاده است و ذخیره سازی تصویر نیز برای مواردی همچون تصویر برداری پزشکی، تصاویر ماهواره ای و نقشه برداری استفاده می شود، بطور کلی اساساً فشرده سازی سیگنال

به منظور حذف همبستگیهای موجود بین داده ها بکار می رود ، در فشرده سازی تصویر با توجه به نوع همبستگیهایی که حذف می شوند فشرده سازی به دو نوع زیر تقسیم می شود :

Intra farme (درون فریمی) : از همبستگی مکانی موجود در هر فریم استفاده می شود .

Inter farme (بین فریمی) : از همبستگی وارتباط بین فریمهای متوالی استفاده می شود .

ولی نوع درون فریمی کاربردی تر است .

۳-۱۳- تاریخچه فشرده سازی اطلاعات [6]

- ۱- تند نویسی(خلاصه نویسی) ۱ قرن قبل از میلاد .
 - ۲- ۱۸۲۹ کشف خط بریل (برای فشرده سازی اطلاعات ارسالی) توسط لوئیس بریل (برای نابینایان) .
 - ۳- ۱۸۳۴ خط مورس برای تلگراف بصورت خط - نقطه وفاصله است .
- با این سه اقدام ایده فشرده سازی مدرن شکل گرفت .
- ۱۹۶۰ تئوری اطلاعات و اجزای آماری ایجاد شد که خود عاملی برای بوجود آوردن کلاسی از الگوریتمهای فشرده سازی با الگوهایی با طولهای متفاوت برای کد کردن مستقل علائم بر پایه تعداد تکرار آنها شد .
- دو روش :

۱- کدینگ هافمن Haffman Coding .

۲- کدینگ محاسباتی که در ۱۹۷۹ توسط ریس (RISS) ارائه شد .

در سال ۱۹۷۷ زیو (ZIV) ودرسال ۱۹۷۸ لمپل (LEMPER) کلاسهای دیگری از الگوریتم فشرده سازی بنام الگوریتمهای دایره المعارفی (Dictionary Algorithms) ایجاد کردند .

همگام با این روشها سعی شد با استفاده از تبدیلات خطی داده ها را به فضای دیگری انتقال دهند که همبستگی بین آنها را به کمترین مقدار برسانند .

۳-۱۴- ساختار کلی سیستم های فشرده سازی اطلاعات شامل بخشهای [8]

۱- کدکننده اطلاعات (شامل نمونه برداری و کوانتیزه کردن داده)

۲- الگوریتم فشرده سازی

۳- ذخیره سازی یا ارسال

۴- الگوریتم بازسازی کدهای فشرده .

۵- باز سازی اطلاعات با استفاده از کدهای باز سازی (دی کدینگ)

۳-۱۵- دی کدینگ

با عمل عکس روند کد کردن به اطلاعات اصلی خواهیم رسید ، ولی اگر داده ها کوانتیزه هم شده باشند در بازسازی کمی خطا داریم .

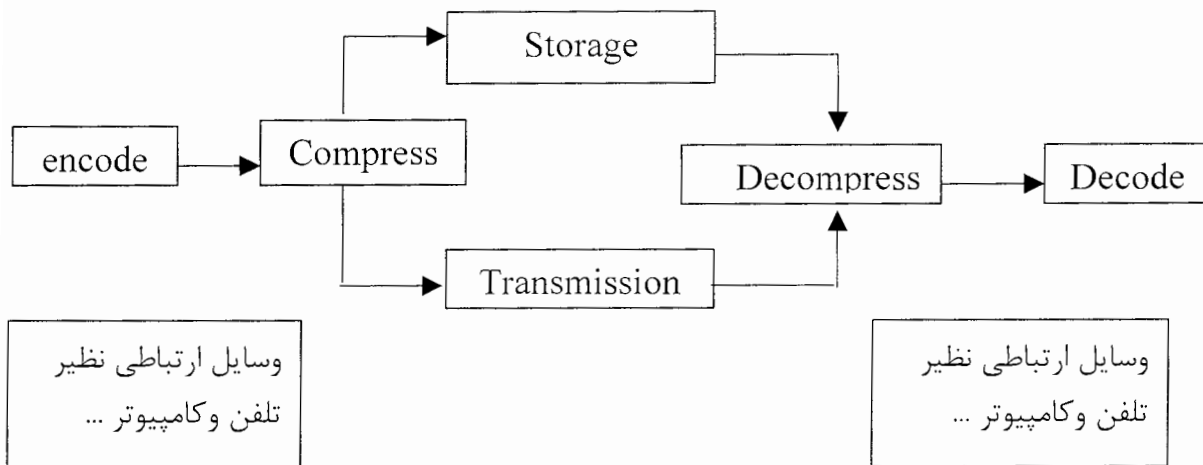
۱- یعنی تبدیل انواع داده (تصویر و صوت و.....) به داده قابل ارسال (اطلاعات باینری) .

۲- در این مرحله ما یکی از الگوریتمهای فشرده سازی را که در ادامه شرح می دهیم ، بکار می بریم .

۳- چون اطلاعات تصویری دارای حجم زیاد است و ما در میزان حافظه و پهنای باند ارسال محدودیت

داریم به فکر استفاده از متدهای فشرده سازی می افتیم . پس با این کار می توان اطلاعات زیادی را با

پهنای باند کم و حافظه کم با سرعتهای بمراتب بالاتر ، ارسال یا ذخیره سازی کرد .



شکل ۳-۱- ساختار سیستم های فشرده سازی اطلاعات

۳-۱۶- روشهای فشرده سازی اطلاعات [6]

روشهای فشرده سازی اطلاعات تصویری :

۱- کدینگ پیش بینی (Predictive Coding)

۲- کدینگ تبدیل (Transform Coding)

در نوع اول با داشتن یک پیکسل پیکسلهای کناری تخمین زده می شوند . از جمله مدولاسیون پهنای پالس تفاضلی (DPCM) و مدولاسیون دلتا .

در نوع دوم ، تصویر با استفاده از تبدیلات خطی به آرایه دیگری تبدیل می شود . بطوری که مقدار زیادی از اطلاعات آن در تعداد کمتری از ضرایب بدست آمده ، متمرکز می شود و روشهای تلفیقی که ترکیبی از دو نوع بالا هستند به کدینگ هایبرد معروفند و می توان آنها را گروه سوم بنامیم . بجز این سه گروه می توان گروه دیگری از روشها را نام برد که به کدینگ پیکسل معروفند ، از این گروه کدینگ طول گام Run-Length Coding مشهور تر است .

۳-۱۷- خلاصه روشهای فشرده سازی :

Image data compression techniques

Pixel Coding :	Predictive Coding	Transform Coding	Other method :
* P CM / quantization	* Delta modulation	* Zonal Coding	* Hybrid Coding
* Run - Length Coding	* Line - by - Line DPCM	* Thershold Coding	* Two - tone / graphics Coding
	* 2 - DPCM		* Color Image Coding

شکل ۳-۲- روشهای فشرده سازی

۳-۱۸- کدینگ طول گام (Run - Length Coding): [6]

ساده ترین روش فشرده سازی است ، برای مثال داده های زیر را در نظر بگیرید :

{ a a b b b c d d d d d e } (۳۵-۳)

برای هر کد دو عدد داده می شود ، که یکی بیانگر طول یا تعداد تکرار و دیگری بیانگر خود الگو است ، پس تبدیل رشته بالا باین صورت است :

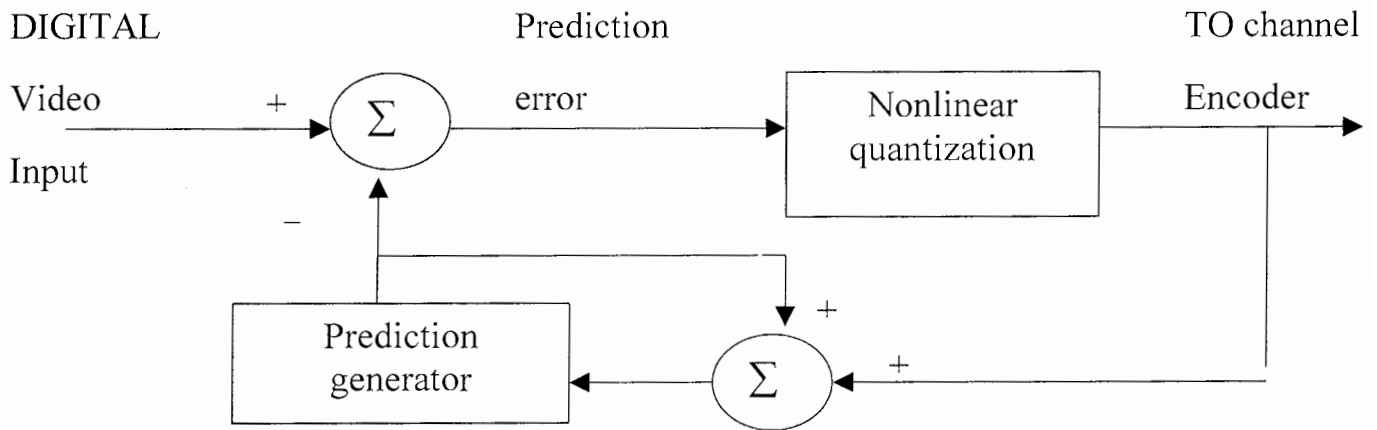
{ 2 a 3 b 1 c 5 d 1 e } (۳۶-۳)

۳-۱۹- کدینگ مدولاسیون پهنای پالس تفاضلی DPCM:

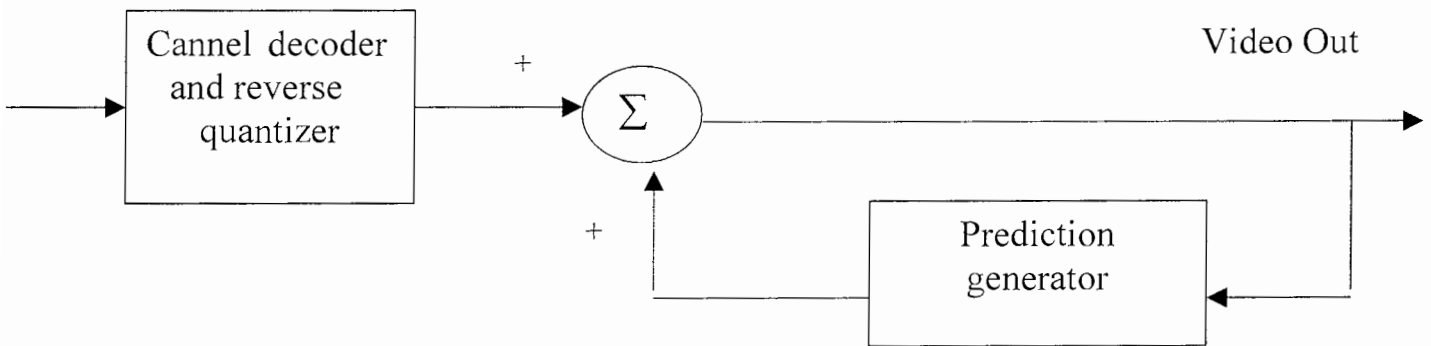
روش توسط کاتر (CUTTER) در سال ۱۹۵۲ ساخته شد . این روش برای کاهش داده در تلویزیونهای دیجیتال مناسب است . و می تواند از یک منبع ۸ بیتی ، تصویری با کیفیت خوب توسط ۴ بیت بر نمونه و تصویری با کیفیت قابل قبول توسط ۳ بیت بر نمونه تولید کند و چون خروجی این کدکننده هنوز دارای تابع چگالی غیر یکنواخت است . می توان از کدینگ آماری استفاده کرد (بطور مثال کدینگ طول گام) و به نرخهای بیشتری در امر فشرده سازی رسید .

۳-۲۰- روش چندی سازی برداری (VECTOR QUANTIZATION): [8]

یکی از قدرتمندترین روشها می باشد. در این روش سعی شده است برای هر مجموعه داده (بردار داده) یک کد داده شود یا بعبارت دیگر در این روش بردارها را اندیس دار می کنند در نتیجه نرخ فشرده سازی بالایی دارد ولی بعلت نیاز به حجم بالای حافظه محاسباتی ، در فشرده سازی تصویر کمتر مورد توجه می باشد .



شکل ۳-۳- بخش کد کردن



شکل ۳-۴- الگوریتم کدینگ DPCM

۳-۲۱- کدینگ تبدیل [6]

فشرده سازی تصویر براساس کدینگ تبدیل بعلت کاربردهای گسترده و نیز کارایی بالا ، دارای اهمیت زیادی است و نیاز به تبدیل فوریه ، هارتلی ، T ، Cas-Cas دارد و نیز این تبدیلهها باید اورتونرمال باشند که هستند .

۳-۲۱-۱- خواص این تبدیلهها :

۳-۲۱-۱-۱- حفظ انرژی (Energy Conservation)

۳-۲۱-۱-۲- تجمع انرژی (Energy Compaction)

۳-۲۱-۱-۳- ناهمبسته کردن داده ها (Data Decorrelation)

۳-۲۱-۱-۱- اگر ماتریس تبدیل ما یک ماتریس بنام A باشد و A ماتریسی اورتونرمال باشد و $U(m,n)$ $V(m,n)$ به ترتیب نشان دهنده تصاویر اصلی و تبدیل یافته باشد در آن صورت با استفاده از رابطه زیر داریم :

$$\sum_{l=0}^{N-1} \sum_{k=0}^{N-1} |U(k, l)|^2 = \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} |V(k, l)|^2 \quad (37-3)$$

این رابطه به مفهوم حفظ انرژی سیگنال ، در تبدیلات اورتونرمال است .

۳-۲۱-۱-۲- در اکثر تبدیلهای یکه کسر بزرگی از متوسط انرژی تصویر در اعضای نسبتاً کمتری از کل ضرایب تبدیل یافته قرار می گیرند . این بدین مفهوم است که با وجود کاهش چشم گیر داده ، انرژی سیگنال تفاوت چندانی با حالت قبل از فشرده سازی ندارد .

۳-۲۱-۱-۳- عموماً داده های یک سیگنال ورودی (مانند یک بلوک $N \times N$ تصویر) دارای همبستگیهای فراوانی نسبت بهم هستند یکی از مهمترین خواص تبدیلات اورتونرمال یکه ، ناهمبسته کردن این داده ها می باشد ، که دقیقاً از این خاصیت برای فشرده سازی سیگنال استفاده می شود .

۳-۲۲- خواص یک تبدیل برای فشرده سازی مؤثر [8]

۳-۲۲-۱- از بین بردن همبستگیها :

معمولاً تغییرات المان‌ها روی تصویر نرم است و شدید نیست، بنابراین می‌توان از وجود همبستگی بین المانها در فشرده‌سازی استفاده کرد. هدف از نگاشت تصویر توسط تبدیلهای گفته شده کاهش همانندی (وابستگی) بین مولفه‌هاست، دراطلاعات تصویری المانها وابستگی زیادی بهم دارند و هر یک اطلاعات مستقلی ندارند و اگر تک تک آنها را کد کنیم راندمان پائینی داریم بدین سبب ابتداء سعی می‌شود این المانها به فضایی نگاشته شود، که مؤلفه‌های مستقل از هم داشته باشیم. سپس مولفه‌ها را بطور منفرد کد می‌کنیم.

یک تبدیل بهینه مثل KLT(Karhunen-leove) وجود دارد که داده‌های داخل تصویر را بطور کامل ناهمبسته می‌کند و باعث فشرده‌سازی بالایی می‌شود، ولی چون حجم زیادی محاسبه لازم دارد نمی‌توان در زمان واقعی از آن استفاده کرد. بعلت وجود این مشکلات و نیز سختی پیدا کردن توابع پایه‌ی زیادی که ناهمبسته باشند، این روش، محاسبات مشکلی خواهد داشت. پس بجای KLT به تبدیل DCT روی می‌آوریم.

۳-۲۲-۲- خطی بودن :

خطی بودن یک تبدیل، نگاشت یک به یک بین المانهای تصویر و ضرایب تبدیل را باعث می‌شود یعنی برای A تبدیل معکوس B را داریم که :

$$B = A^{-1} \quad (3-38)$$

۳-۲۲-۳- اور تو گونالیتی :

وقتی که از یک تبدیل استفاده می‌کنیم باید انرژی در هر دو دامنه یکسان باشد و هیچ انرژی از دست نرفته باشد و یا اینکه تبدیل ما باعث ایجاد اطلاعات زاید نشود.

۳-۲۲-۴- عملکرد سریع :

سرعت عملکرد یک تبدیل باید بالا باشد تا بشود از آن استفاده کرد. البته برای هر تبدیل الگوریتم - های سریعی ارائه شده است که این مشکل را حل می‌کند. مثلاً برای تبدیل N نقطه ای یک بعدی در ابتداء N^2 عملیات لازم است که با روشهای الگوریتم سریع به مقدار $N \log N$ کاهش می‌یابد و در دو بعدی N^4 عملیات لازم است که آن هم به $2N^2 \log N$ عملیات کاهش می‌یابد.

۳-۲۳- انتخاب تبدیل مناسب [6]

همانطور که اشاره شد تبدیل KLT تنها تبدیلی است که خواص بالا را بصورت عالی داراست و ماتریس تبدیلیش از قطری کردن ماتریس کواریانس داده های تصویری بدست می آید که ضرایب تبدیل کاملاً غیر همبسته خواهند شد. از آنجائیکه واریانس های ضریب تبدیل KLT به سمتی میل می کنند که مقدار آنها با افزایش درجه کاهش می یابد در مقایسه با دیگر تبدیلات بهترین فشرده سازی انرژی را ارائه می دهد و یک تبدیل بهینه است ولی بعلت داشتن چهار گام محاسباتی پیچیده زیر:

۱- بدست آوردن ماتریس همبستگی اولیه.

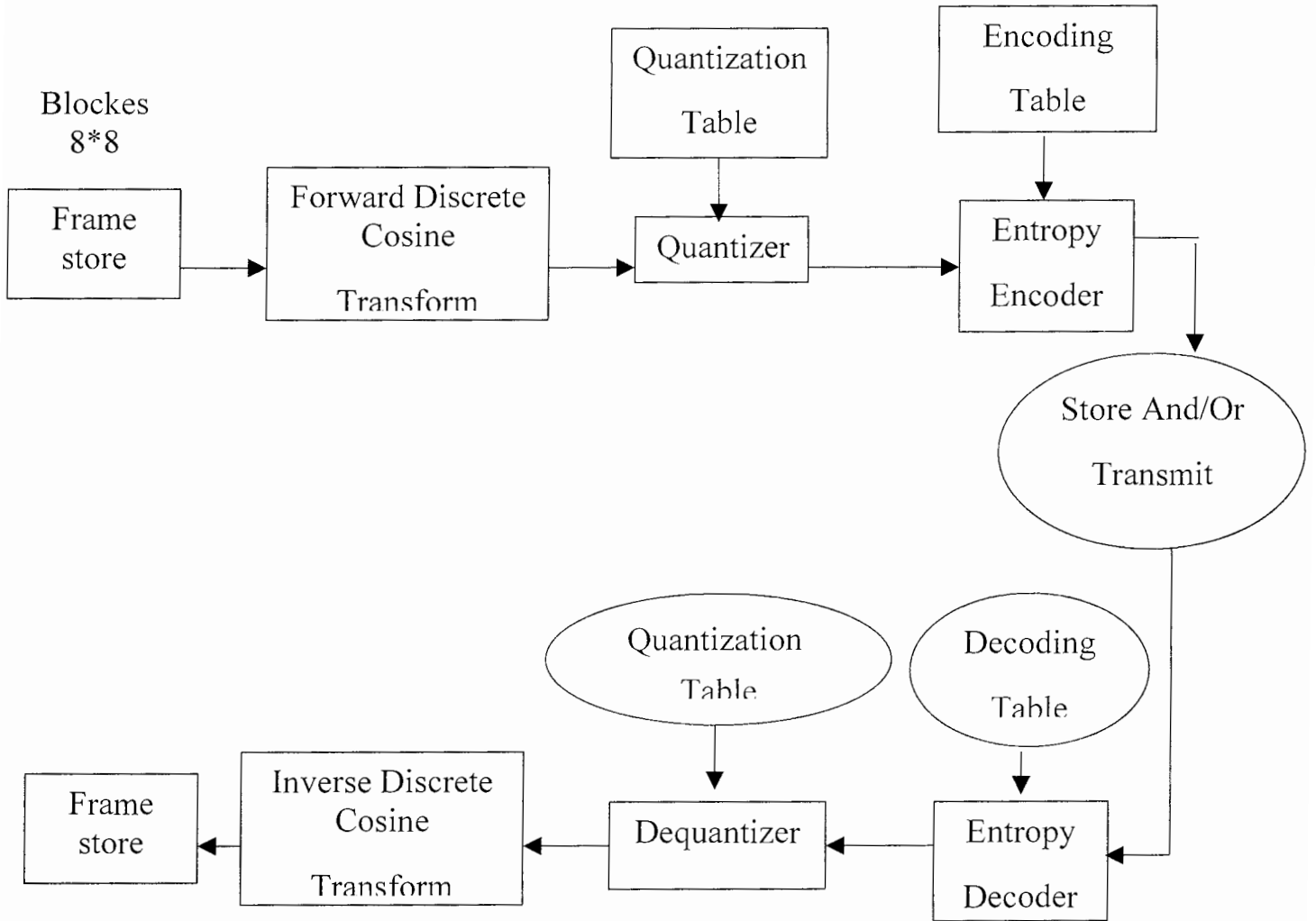
۲- قطری سازی آن.

۳- محاسبه المانهای ماتریس KLT.

۴- محاسبه مقدار هر ضریب تبدیل.

و همچنین چون تصویر بطور عموم در حال تغییر است (غیر ایستاست)، در هر لحظه ماتریس تبدیل معکوس جدیدی مورد نیاز است تا به گیرنده ارسال شود. در همین اثنا هم تصویر جدیدی آماده ارسال می شود و در نتیجه انجام این کار با این روش را غیر عملی می سازد، ولی تبدیلات دیگر، شاید زیر بهینه (Suboptimal) باشند، اما این مشکلات را برطرف می سازند. یعنی چون مجموعه ثابت بردارهای پایه را استفاده می کنند در نتیجه سرعت محاسبات ضرایب تبدیل بالا می رود و نکته در اینجاست که راندمان فشرده سازی تفاوت زیادی هم با روش KLT ندارد، بعنوان مثال تبدیلات DHT, DCT, FFT.

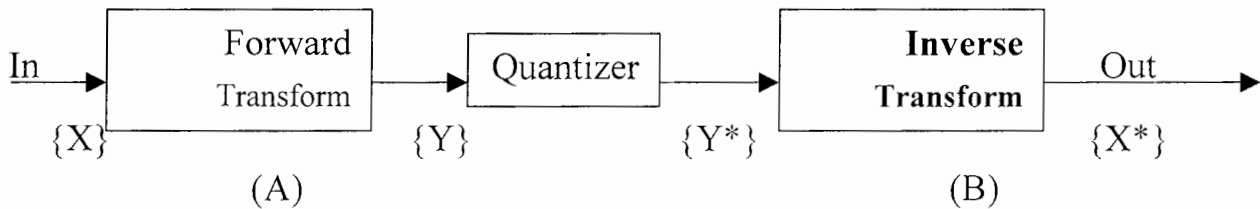
تبدیل DCT یک تبدیل حقیقی و نزدیکترین تقریب به KLT است و بطور کارایی در فشرده سازی تصویر استفاده می شود. الگوریتمهای مشهوری، از جمله JPEG، که براساس تبدیل DCT نوشته شده است، را می توان نام برد.



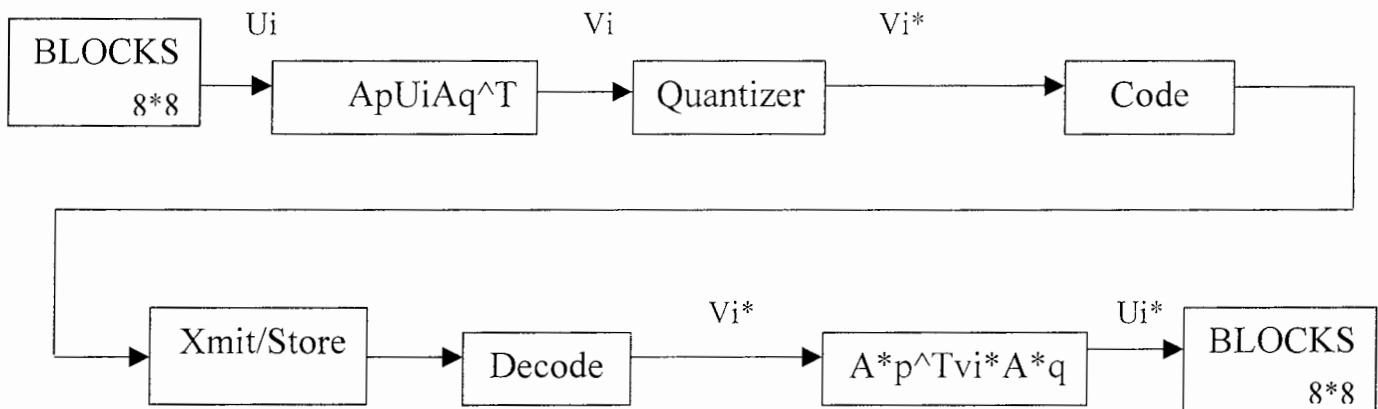
شکل ۳-۵- ساختار کدینگ و بازسازی JPEG

از آنجاییکه DHT نیز یک تبدیل حقیقی است و جمله اضافی سینوس را نسبت به DCT دارد، شایسته است که خصوصیات آن را در مورد فشرده‌سازی بررسی کنیم. می‌دانیم که تبدیلات DHT و FFT به یکدیگر قابل تبدیل هستند.

۳-۲۴- ساختار الگوریتمهای کدینگ یک بعدی و دو بعدی :



شکل ۳-۶- ساختار کدینگ یک بعدی



شکل ۳-۷- ساختار کدینگ دو بعدی

در کدینگ یک بعدی که رشته ورودی $\{X_n\}$ و $n=0 \dots N-1$ تحت تبدیل A به $\{Y_n\}$ تبدیل می‌شود و $\{Y_n\}$ شامل N ضریب تبدیل می‌باشد ضرایب بطور جداگانه کوانتیزه می‌شوند با این عمل امکان کاهش داده یا فشرده‌سازی آن را فراهم می‌آورد. در گیرنده ضرایب کوانتیزه شده $\{Y_n^*\}$ با استفاده از تبدیل $B = A^{-1}$ به $\{X_n^*\}$ تبدیل می‌شوند که تخمینی از $\{X_n\}$ است. [6]

۳-۲۵- تبدیل کسینوس گسسته (DCT): [6]

بعد از KLT بهترین تبدیل برای فشرده سازی DCT است. در آزمایشات مختلف نشان داده شده است که DCT برای رشته مارکوف ایستای مرتبه اول با $p > 0.5$ (ضریب همبستگی مرتبه اول) نزدیکترین کارایی را به KLT دارد.

ماتریس کسینوس $N \times N$ و $C = \{c(k,n)\}$ که آنرا DCT می نامند، بصورت زیر تعریف می شود:

$$C(k,n) = \begin{cases} \frac{1}{\sqrt{N}} & k=0, 0 \leq n \leq N-1 \\ \sqrt{\left(\frac{2}{N}\right)} \times \cos\left[\frac{\pi(2n+1)k}{2N}\right] & \begin{cases} 1 \leq k \leq N-1 \\ 0 \leq n \leq N-1 \end{cases} \end{cases} \quad (3-39)$$

DCT یک بعدی رشته $\{u(n), 0 \leq n \leq N-1\}$ بصورت زیر خواهد بود:

$$V(k) = \alpha(k) \sum_n U(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], 0 \leq k \leq N-1 \quad (3-40)$$

که در آن:

$$\alpha(k) = \sqrt{\left(\frac{2}{N}\right)}, \alpha(0) = \sqrt{\left(\frac{1}{N}\right)} \quad (3-41)$$

است و تبدیل معکوس آن:

$$U(n) = \sum_k \alpha(k) V(k) \cos\left[\frac{\pi(2n+1)k}{2N}\right], 0 \leq n \leq N-1 \quad (3-42)$$

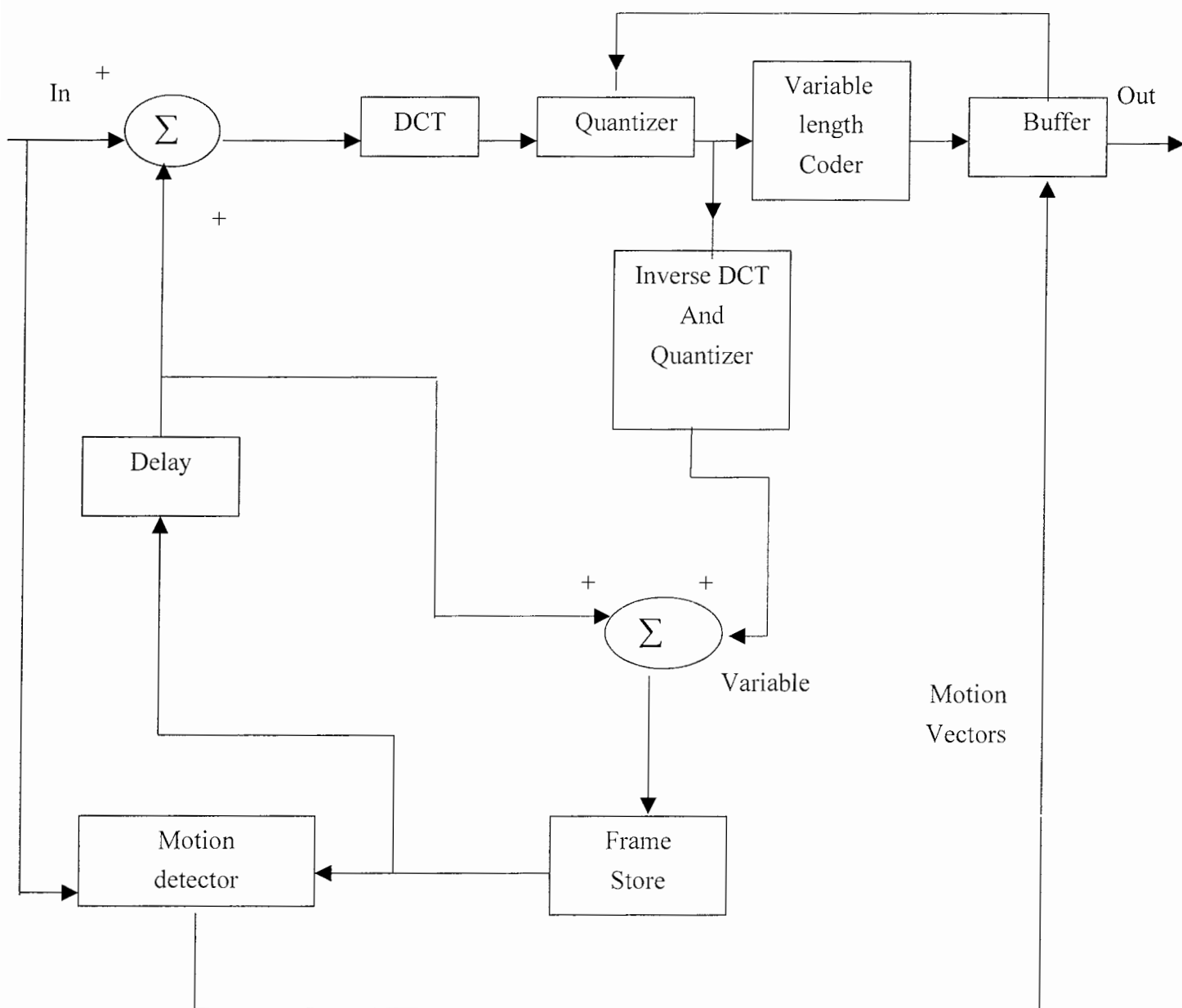
می باشد.

DCT اکثر انرژی را در چند ضریب تبدیل نگه می دارد. هر چه داده های تصویری دارای همبستگی بیشتری باشند، تجمع انرژی بیشتر است، غیر از آن روشهای سریعی هم برای محاسبه دارد.

۳-۲۶- کدینگ تلفیقی

بطور کلی هر گاه از چند روش فشرده سازی باهم استفاده کنیم به آن کدینگ تلفیقی گویند. برای

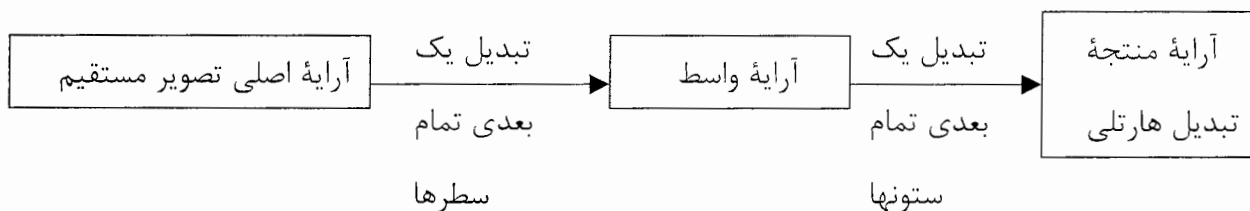
مثال تلفیق دو روش DCT, DPCM.

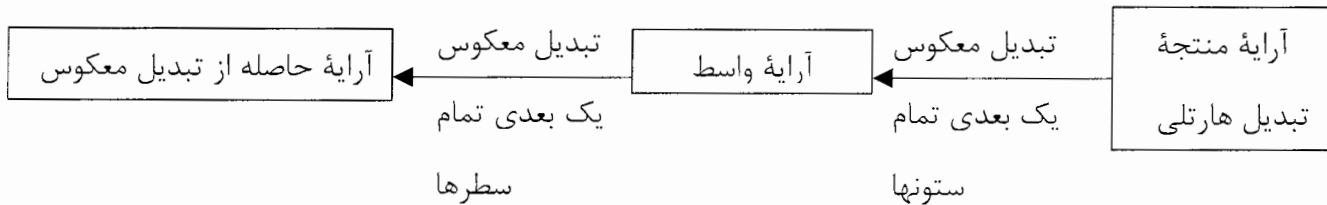


شکل ۳-۸ - تلفیق دو روش DCT, DPCM

۳-۲۷ - تبدیل دو بعدی هارتلی [7]

به این صورت است که چون خاصیت جدایی پذیری به تبدیل یک بعدی را دارد ابتداء تمام سطرها و سپس تمام ستونها را تبدیل می کنیم تا آرایه تبدیل هارتلی بدست آید و برای معکوسش هم عکس عمل بالا (ابتداء تمام ستونها را تبدیل معکوس کرده و سپس تمام سطرها را تبدیل معکوس می کنیم):





شکل ۳-۹- ساختار تبدیل و تبدیل معکوس دوبعدی هارتلی

۳-۲۸- چگونگی اعمال یک تبدیل به تصویر

معمولاً کدینگ تبدیل بر روی تمام تصویر یک مرتبه انجام نمی گیرد، بلکه بر روی زیر بلوکهای ثابتی (معمولاً $8*8$ و $16*16$) انجام می گیرد. علت این تقسیم بندی:

- ۱- محاسبهٔ تبدیل یافتهٔ بلوکهای کوچکتر ساده تر از محاسبهٔ تبدیل یافتهٔ کل تصویر است.
 - ۲- حجم حافظهٔ مورد نیاز برای محاسبهٔ تبدیل تصویر کاهش می یابد.
 - ۳- همبستگی پیکسلها برای پیکسلهای دور از هم کمتر از پیکسلهای مجاور است.
- اگر تصویر دارای اندازه $M*N$ باشد و زیر تصویرها را $p*q$ در نظر بگیریم، با تقسیم تصویر به زیر تصویرهای کوچکتر علاوه بر کاهش حجم حافظهٔ اصلی برای پیاده سازی تبدیل، به نسبت $(MN)/(pq)$ بار، محاسبات نیز به نسبت $\log(MN)/\log(pq)$ در مبنای دو کاهش می یابد. هرچه اندازهٔ بلوکها بزرگتر شود نرخ بدست آمده برای فشرده سازی نیز کاهش می یابد، اما با کاهش اندازهٔ زیربلوکها پیچیدگی سخت افزاری الگوریتم افزایش می یابد، که مقدار Typical ما همان $8*8$ می باشد.

۳-۲۹- چگونگی حذف ضرایب در تصویر تبدیل یافته برای فشرده سازی تصویر

برای اینکه داده های تصویری را فشرده کنیم باید در تصویر تبدیل یافته ضرایبی را که از اهمیت کمتری برخوردارند حذف کنیم، که معمولاً این ضرایب همان فرکانسهای بالای تصویر است. چون می دانیم که بیشتر انرژی تصویر در فرکانس های پائین است.

تعداد ضرایبی را که مجاز به حذف آنها هستیم ثابت نیست و وابسته به نوع تصویر تغییر می کند و حذف ضرایب تا وقتی تصویر قابل قبول بماند، ادامه پیدا می کند.

۳-۳۰- خطای بازسازی [4]

برای بدست آوردن تصویر خطا (Error Image) کافیهست که تصویر اصلی (True Image) را از تصویر بازسازی شده (Restored Image) کم کنیم :

$$E \text{ Image} = R \text{ Image} - T \text{ Image}$$

برای آنکه بتوانیم خطای بازسازی را با عدد نشان دهیم از میانگین مربعات خطا (MSE) استفاده می کنیم. که طبق انتظارمان خطاها بصورت زیر است :

$$\text{ERROR(DCT)} < \text{ERROR(FHT)} < \text{ERROR(FFT)}$$

پس نتیجه می گیریم که خطای روش DCT از همه روشها کمتر است و بعد از آن روش FHT مقام دوم را دارد .

طرز عملکرد ، روش DCT را بعنوان مثال درضمیمه (۲) با نرم افزار مطلب ، خواهید دید .

۳-۳۱- کاربردهای دیگر تبدیل هارتلی :

۱- در فشرده سازی صوت .

۲- در بازشناسی تصویر و بازشناسی صوت .

۳- برای حذف نویز و بهبود تصاویر .

۴- شناسایی سیستم .

بخش دوم

کنترل ترافیک

فصل (٤)

مرور کارهای انجام شده تا کنون

۴-۱- مقدمه

امروزه ، فواید حاصله از کنترل ترافیک دیگر امری سئوال برانگیز نیست . مطابق با این امر برای بکار بردن الگوریتم های کنترلی حلقه بسته ، برای ایجاد توانایی تطبیق با شرایط متفاوت ، نیاز به یک سنسور برای اندازه گیری جریان حرکتی وسایل نقلیه داریم . چندین نوع سنسور تا به حال بر اساس اصول مختلف عملکرد ، از جمله : انتشار امواج آلتراسونیک ، تونل های باد و حلقه های مغناطیسی طراحی شده اند . باوجود این همه ، این سنسورها مشکلات و موانع گوناگونی را از جمله : قیمت بالای نصب و سخت بودن عملیات نصب و یا عدم دقت و درستی تحت شرایط مختلف ترافیکی را ایجاد کرده اند . اما سنسور هایی که بر اساس روش های پردازش تصویر کار می کنند ، ما را به سمت خودشان جذب می کنند . [11]

ما با مطالعه در این زمینه به سه مشکل اصلی بر خوردیم :

۱- اولین آنها از درخواست برای عملیات های پردازشی همزمان سرچشمه می گیرد . به عنوان نمونه حجم حافظه مورد نیاز برای عملیات در زمان واقعی اگر تصویر دارای ابعاد ۵۱۲×۵۱۲ پیکسل باشد و ما با ۳۰ فریم در ثانیه فیلم برداری کرده باشیم ، پس نرخ داده : $۷۸۶۴۳۲۰ \times (۵۱۲ \times ۵۱۲ \times ۳۰)$ بایت بر ثانیه و یا معادل بیش از ۶۰ مگابیت بر ثانیه است . که آن خیلی بیشتر از مقادیر قابل ارسال با تکنولوژی های حال حاضر است .

۲- مشکل دوم از مختلف بودن الگوریتم های تحلیلی برای تصویر ایجاد می شود ، که آن نیز از گوناگونی شرایطی که برای تصویر پیش می آید ، از جمله : بارندگی ، مه ، سایه ، بازتاب ، تغییر روشنایی تصویر و نیز تفاوت شکل خودروها ، ناشی می شود .

این دو مشکل با توجه به این که همه تصاویر نیاز به تحلیل ندارند ، کمرنگ می شوند . برای مثال اگر فرض کنیم که ماشینی سرعت ۱۰۰ کیلومتر در ساعت رادارد و طولش ۴ متر است ، اگر دوربین ما ۱۰۰ متر از خیابان را پوشش دهد و با سرعت ۱۰ فریم در ثانیه نیز فیلم بگیرد ما ۳۶ عکس از خودرو راداریم که برای آشکارسازی آن کافی می باشد . در ضمن ما نمی خواهیم کل تصویر را پردازش کنیم ، بلکه تنها بخش مورد نظر ROI (Region Of Interest) برای ما مهم است .

۳- مشکل سوم و آخرفاکتور غیر اقتصادی بودن است. روش های پردازش تصویر نیازمند به : سخت افزار برای یک سیستم تصویری، یک دوربین ویدیویی مخصوص و یک کامپیوتر سریع دارند، که در مقایسه با روش های مشهور و سنتی حلقه های کنترلی قیمت خیلی بالایی دارند و گران تمام می-شوند . باوجود این ، با توسعه کامپیوترها و ابزارهای ویدیویی و تصویری هزینه این عناصر سازنده سیستم به شدت کاهش پیدا کرده است .

دو نکته اصلی درمورد مطالعه در زمینه کنترل ترافیک مورد توجه است. اول کمک گرفتن از الگوریتم هایی که تنها قسمتی از تصویر را مورد استفاده قرار می دهند و دوم بکار بردن الگوریتم هایی که کل تصویر را مورد استفاده قرار می دهند. شایان ذکر است که ما روش اول را برای کم کردن مشکلات روش های پردازش تصویر را انتخاب کردیم . بر طبق این دو نظریه : هیچ سیستمی نیست که بتواند بر تمام ناملایمات غلبه کند ، از جمله : اعوجاج های ناشی از فیلم برداری ، تغییر شرایط آب و هوایی، انعکاس ، سایه و روشنی تصویر ، اشکال مختلف ماشین ها و زمان پردازش و.... [12]

به نظر می رسد که این مشکلات تا به حال به صورت مقطعی برای هر سیستم حل شده است و این موضوع هنوز باز مانده است. این مطلب ناشی از چشم داشت های مالی برای این سیستم هاست که باعث شده است که اغلب اطلاعات و نتایج تازه و اخیر در مورد این موضوع مخفی باقی بماند .

۴-۲- کنترل ترافیک به روش موزاییک [15] (Mosaik Method)

روش موزاییک که با گرفتن مربع هایی روی قسمت های متحرک شکل کار می کند ، یعنی به هربخش متحرک تصویر یک مربع اختصاص می دهد .

چند نکته در این روش حائز اهمیت است :

۱- زمان رسیدگی به تغییر در تصویرهای جدید برای رسیدگی به کنترل مهم است .

۲- خطاها مقدار قابل قبولی را دارند .

دراین روش ابتدا تصویر به مناطقی که حرکت یکسانی دارند ، تقسیم می شود . نواحی که

حرکت یکسانی دارند $R_1 \dots R_n$ نامیده می شوند و

R: تشخیص جزء تصویر

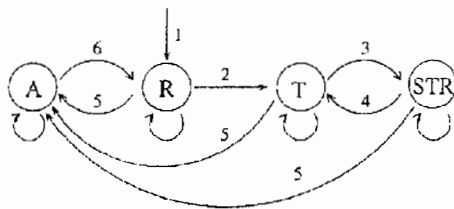
T: دنبال کردن سریع

STR: دنبال کردن و تشخیص همزمان

A: رسیدگی و کنترل

می باشند . سپس تصمیم می گیریم که آیا هر بخش جزء کار هست یا نه؟ البته بر اساس تکنیک

مشاهدهء گروهی این امر محقق می شود . مطابق با این الگو:

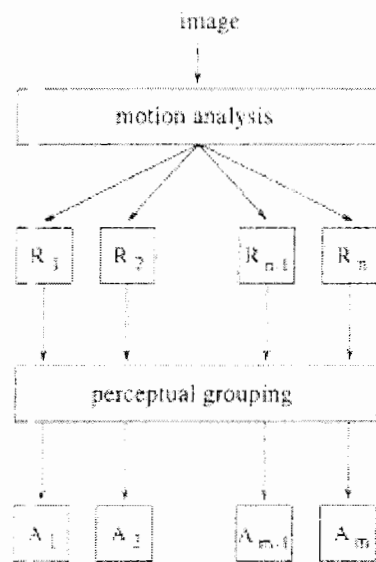


شکل ۴-۱ مراحل الگوریتم Mosaik

حال که تصویر به مناطقی مربع شکل که حرکت یکسانی دارند ، تقسیم شد ($R_1 \dots R_n$)، باید برای

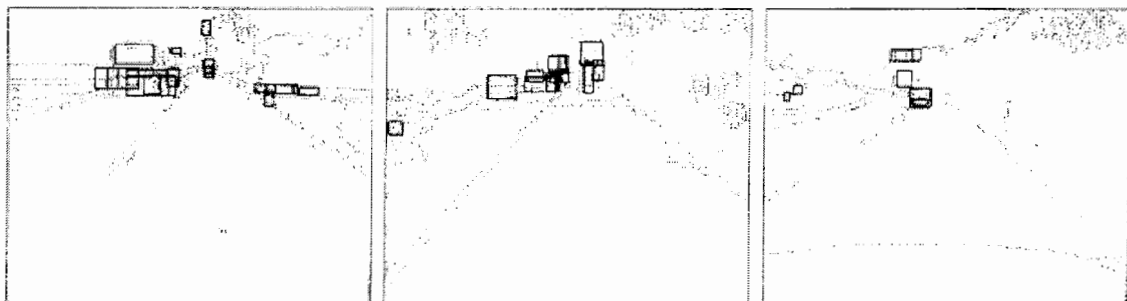
کاهش حجم داده های تصویری قسمت های متحرک آن را به روش آنالیز حرکت جدا کنیم وباتکنیک

مشاهدهء گروهی آن را به $(A_1 \dots A_m)$ که $m < n$ است تبدیل کنیم ، مطابق با این الگو :



شکل ۲-۴ روند تقسیم بندی تصویر و انتخاب نواحی متحرک

خلاصه این روش اینست که ابتدا تصویر به n قسمت با حرکت یکسان تقسیم می شود و بعد از آن به روش مشاهدهء گروهی قسمت های متحرکش جدا می شود. بعد روی هر قسمت یک مربع می-اندازد و آنها را شمارش می کند. مطابق با شکل زیر:



شکل ۳-۴ طرز کار روش Mosaik بر روی تصاویری از ترافیک

۴-۳- کنترل ترافیک به روش تخمین سرعت وسایل نقلیه [16]

هدف از کنترل ترافیک حداکثر استفاده از ظرفیت جاده و حداقل کردن میزان معطلی در جاده است. الگوریتمی که در این قسمت بر اساس پردازش تصویر شرح داده می شود بر پایه تشخیص سرعت خودروها استوار است. مراحل انجام این روش بدین شرح است:

۱- محاسبه سرعت فریم ها در امر فیلم برداری (fps) و محاسبه طولی از جاده که تحت پوشش دوربین است. (L)

۲- Initial zone یا در واقع ناحیه اولیه ای که خودرو وارد میدان دید دوربین می شود، برنامه آنقدر در این مرحله می ماند تا اینکه حرکتی آغاز شود.

۳- وقتی حرکتی آغاز شد، شماره فریم را در Initial frame ثبت می کند.

۴- برنامه آنقدر در این مرحله می ماند تا اینکه حرکتی در ناحیه Final zone یا همان ناحیه پایانی دیده شود.

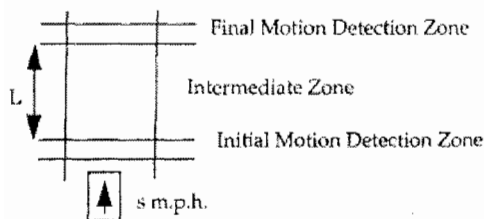
۵- به محض مشاهده حرکت در ناحیه پایانی شماره فریم را در Final frame ثبت می کند.

۶- در نهایت با فرمول زیر سرعت خودروها را تخمین زده و کنترل های لازم را اعمال می کند.

سرعت متوسط از فرمول:

$$s = \frac{L}{t_2 - t_1} \quad (1-4)$$

حاصل شده است و در نهایت طبق شکل روبرو:



شکل ۴-۴ روش تخمین سرعت خودرو

مقدار سرعت برای اعمال بر کنترلرهای موجود از رابطه صفحه بعد حاصل می شود :

$$Speed = \frac{FrameRate \times RoadLength}{FinalFrameNumber - InitialFrameNumber} \quad (2-4)$$

رابطه مربوط به محاسبه خطای نسبی در این روش بصورت :

$$\frac{\Delta s}{s} \times 100 \leq \frac{293 \times s}{f \times L} \quad (3-4)$$

می باشد . همان طوری که مشهود است ، برای داشتن خطای نسبی کمتر باید مقدار (fps) یا همان f و نیز مقدار L را افزایش داد .

۴-۳-۱- بحث

از اشکالات این روش ، وابستگی شدید میزان خطا به fps را می توان نام برد . یعنی ما از طرفی محدودیت سرعت فریم را برای آنالیز داده های ورودی داریم و از طرف دیگر مشکل زیاد شدن خطا ، که این خود بر میزان درستی کار ما اثر می گذارد .

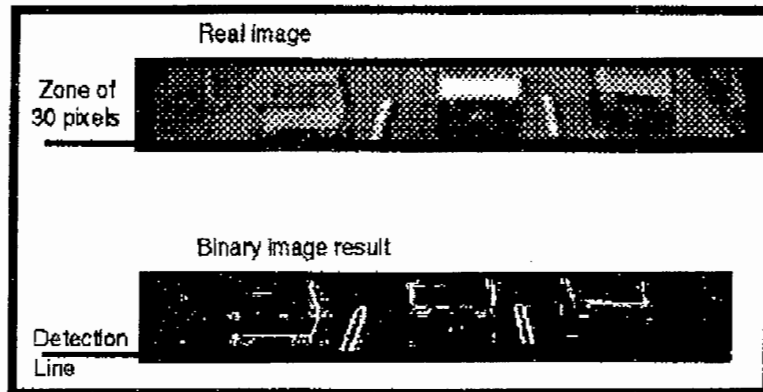
۴-۴- روش های دیگر [13]

در این قسمت ما برای طولانی نشدن بحث دیگر دست به توضیح در مورد روش ها نمی زنیم ، بلکه تنها به معرفی اجمالی آن ها می پردازیم . روش های دیگری از جمله : روش گرادیان (مشتق گیری) : برای پیدا کردن لبه های تصویر ، روش فیلتر کردن با توابع چی بی شف (chebyshev) : برای کاهش میزان نویز و نیز بهبود کیفیت تصویر بعد از دیجیتال شدن ، روش مطابقت دادن با الگو - های ماشین های مختلف : در این روش بعد از شناخت اجزاء متحرک شکل با قرار دادن قالب هایی سعی می کنند که علاوه بر شمارش خودروها نوع آن ها را تشخیص دهند و از نظر آماری تعداد هر نوع از خودروها را مشخص نمایند . به عنوان مثال:

در روش گرادیان گیری از تصویر ورودی ، لبه های تصویر (محل های تغییر سطح خاکستری تصویر) در شکل نهایی مشخص می شوند و از آن می توان برای آشکار سازی خودروها استفاده کرد :

$$G(i, j) = |F(i, j) - F(i, j-1)| + |F(i, j) - F(i-1, j)|$$

که در آن معرف شکل ورودی و $G(i,j)$ معرف شکل بعد از گرادیان گیری است. شکل زیر نمونه ای عملی از طرز کار این روش است :



شکل ۴-۵ آشکارسازی لبه های تصویر را با روش گرادیان نشان می دهد .

برای از بین بردن نویزها و بالا بردن کیفیت تصویر بعد از دیجیتال کردن آن ، می توان از توابعی نظیر این تابع استفاده کرد وبا اعمال فیلتر های بالا گذر باز هم لبه های تصویر را آشکار کرد و سپس تعداد خودروها را شمرد: [13]

$$H(z) = \frac{0.0016 + 0.0049Z^{-1} + 0.0049Z^{-2} - 0.0016Z^{-3}}{1 - 2.5435Z^{-1} + 2.2053Z^{-2} - 0.6483Z^{-3}} \quad (4-4)$$

برای بیان منظورمان در روش الگو گذاری بر قسمت های متحرک شکل و سپس شمارش تعداد هرنوع خودرو به بیان یک نمونه کار با این روش می پردازیم :

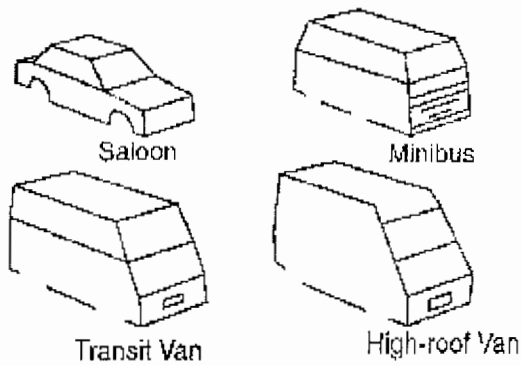
ابتدا انواع خودروها را از نظر ابعاد دسته بندی می کنیم ، مطابق جدول جدول ۴-۱ انواع خودرو

motorcycle
passenger car
pick up
mini van
tractor with trailer
delivery truck
bus
garbage truck
cargo van
recreation vehicle
van
tractor
dump truck
sport utility
other

روبرو :

سپس طبق شکل صفحه بعد ، الگوهای متفاوتی از خودروها را به

برنامه تشخیص الگو معرفی می کنیم :



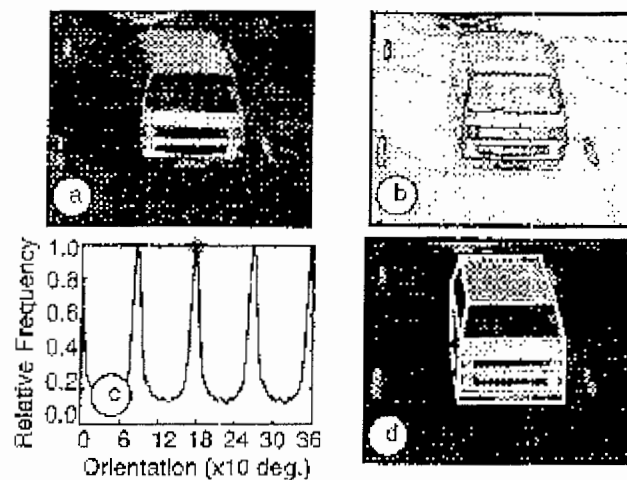
با انجام این کار این برنامه روی قسمت های متحرک شکل عمل تطبیق را انجام می دهد و هر الگو که به تعداد بیشتری ، پیکسل های متحرک شیء را پوشش دهد، آن را به عنوان الگوی آن بخش معرفی می کند و نوعش را تشخیص

می دهد. [17] شکل ۴-۶ چهار مدل متفاوت برای خودروها را به برنامه معرفی می کند.

خروجی برنامه و طرز کار آن در شکل های زیر آمده است :



شکل ۴-۷ نمونه ای از طرز کار روش الگو گذاری (Mask)



شکل ۴-۸ نمونه ای دیگر از طرز کار روش الگو گذاری (Mask)

تنها یک روش دیگر برای معرفی شدن باقی مانده است، که چون از تکنیک خیلی جالبی استفاده کرده است، جداگانه به معرفی آن می پردازیم:

۴-۵- روش Tracking & Speed estimating [19]

این روش برای شمردن ماشین های یک خیابان ابتدا با یک روال (سلسله مراتب) در یک بازه زمانی، تصاویری از یک جاده را گرفته و مجموع ماشین هایی که از جاده مورد نظر عبور کرده اند را حساب می کند. محاسبه تعداد ماشین ها به روش دنبال کردن آن ها در ناحیه مخصوص تعقیب یعنی Tracking-zone صورت می گیرد. این عمل در طی مراحل زیر انجام می شود:

۱- گرفتن تصویری از اصل جاده بدون ماشین

۲- آشکارسازی وسیله نقلیه

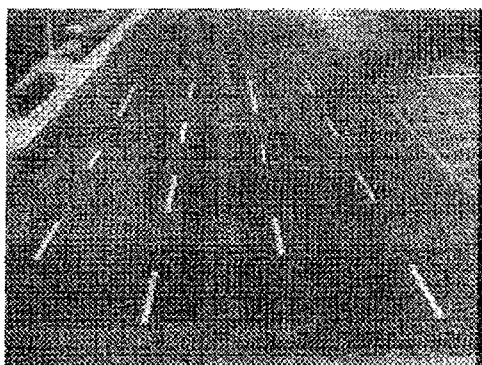
۳- محاسبه سرعت وسیله نقلیه

۴- دنبال کردن حرکت وسیله نقلیه

کار سیستم به آشنایی با شرایط و ساختار جاده و سپس آشکارسازی وسیله نقلیه و دنبال کردن آن با توجه به تخمینی که از سرعتش زده است، در ناحیه Tracking-zone محدود و خلاصه می شود. ورودی سیستم، تصاویری از یک ترافیک است و هدف تعقیب خودروها برای امر شمارش و در نهایت کنترل ترافیک است.

در مرحله نخست یا همان استخراج ساختار جاده (Road Structure Extraction) سیستم،

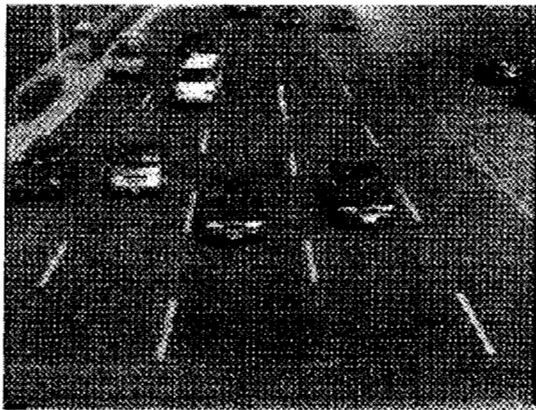
تصویری از اصل جاده بدون ماشین را دریافت می کند. مطابق شکل زیر:



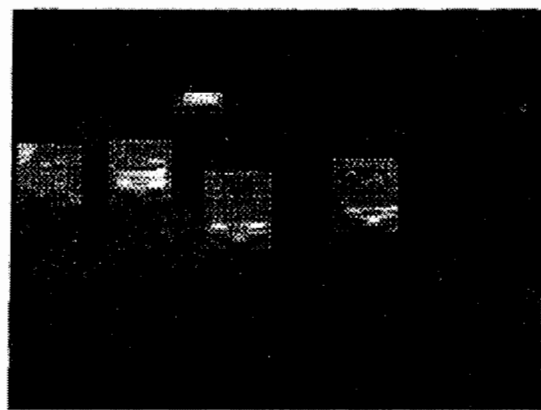
شکل ۴-۹ مرحله استخراج ساختار جاده

۴-۵-۱- روش آشکارسازی

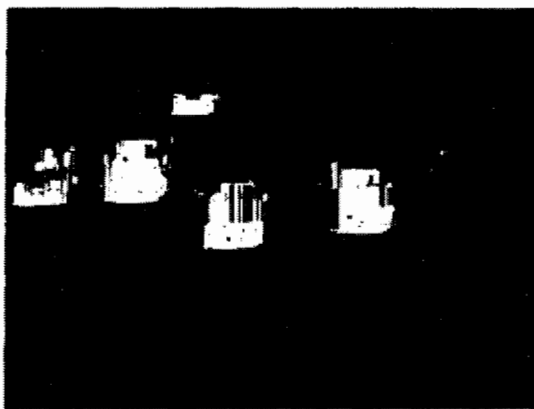
می دانیم که تعداد لبه های افقی در خودرو بیشتر از لبه های عمودی است، که این روش این نکته را به عنوان مبنای آشکارسازی معرفی می کند. برای حفظ لبه های افقی ماشین ها از عملگر Prewitt استفاده می کند. با این کار خطوطی مقطع سفید جاده که عمودی اند حذف می شوند. بعد به روش Run-length Smearing یا همان روش آغشته کردن و ترکیب خطوط افقی، آن ها را به هم می چسباند تا نواحی که به عنوان محل های ماشین ها حدث زده شده اند، مشخص گردند و سپس قالب یک ماشین متوسط، جایگزین این محل ها می شود. مطابق شکل های زیر: [14][18]



شکل ۴-۱۰ تصویر جاده با ماشین



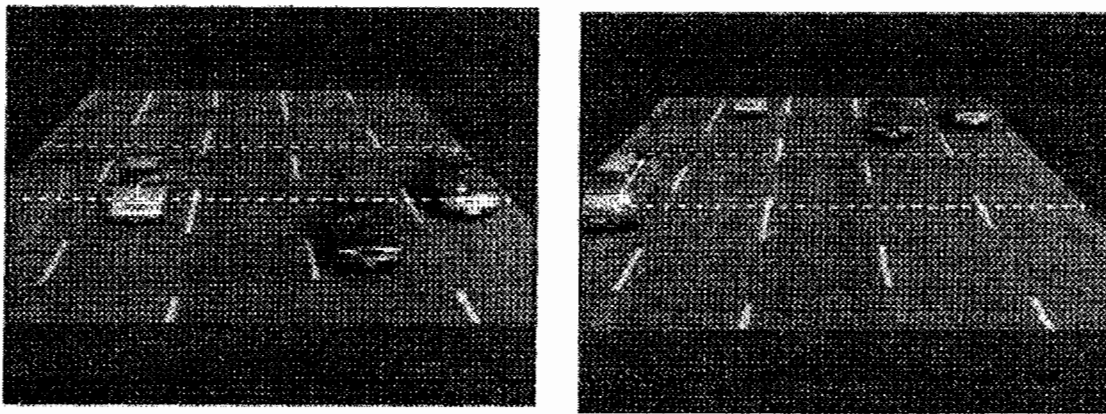
شکل ۴-۱۲ تصویر نهایی جاده با ماشین های آشکار شده



شکل ۴-۱۱ آغشته کردن لبه های افقی تصویر

۴-۵-۲- روش محاسبه سرعت

اگر بخواهیم هر ماشین را که وارد این ناحیه می شود از لحظه ورود تا لحظه خروج تعقیب کنیم، باید کلیه فریم های اندازه ماشین ها را حفظ کنیم که اگر چند ماشین در تصویر باشد، این کار به زمان و حافظه بالایی نیاز دارد. برای این کار این سیستم به جای آن که تمام پیکسل های یک ماشین را بگیرد، یک محدوده کمی از پیکسل های مربوط به آن ناحیه را ذخیره می کند و برای هر ماشین با سرعت بالا بین دو ناحیه از فریم این کار را انجام می دهد و سرعت را تخمین می زند. [19]



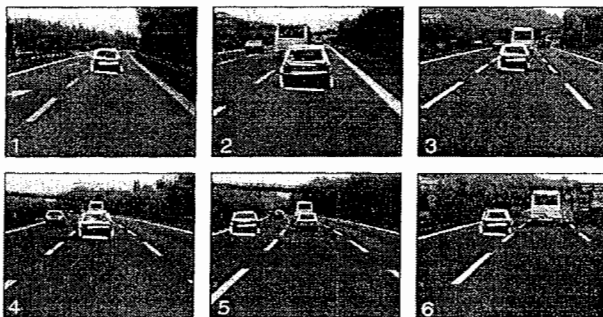
شکل ۴-۱۳ طرز تعقیب در ناحیه خط چین برای تخمین سرعت

(خروج از ناحیه)

(ورود به ناحیه)

در نهایت با ترکیب دو عمل تعقیب و تخمین سرعت، کنترل های لازم برای ترافیک انجام می-

شود. اشکال زیر نمونه دیگری از طرز عملکرد این روش اند: [19]



شکل ۴-۱۵ طرز تعقیب و تخمین سرعت

فصل (۵)

معرفی روش جدید و پیشنهادی پایان نامه

۵-۱- معرفی روش جدید و پیشنهادی پایان نامه

در این قسمت، ما یک سیستم بینایی کامپیوتری را برای شمارش ماشینهای روی جاده ها و خیابان ها مبتنی بر روش های پردازش تصویر دیجیتال توصیف کردیم . تصاویر ترافیکی توسط یک دوربین فیلم برداری گرفته می شوند و بعد از دیجیتال شدن وارد کامپیوتر می گردند. الگوریتم شمارش بر اساس مقایسه بین یک مرجع محاسبه شده بدون ماشین ها و تصویرجاری از خطوط ترافیکی بنا شده است . امتحان این روش تحت شرایط واقعی رضایت بخش بوده است و با این روش بالای ۹۰٪ درستی کار داریم .

ما با مطالعه در این زمینه به سه مشکل اصلی بر خوردیم :

۱- اولین آنها از درخواست برای عملیات های پردازشی همزمان سرچشمه می گیرد . به عنوان نمونه حجم حافظه مورد نیاز برای عملیات در زمان واقعی اگر تصویردارای ابعاد $۵۱۲ * ۵۱۲$ پیکسل باشد و ما با ۳۰ فریم در ثانیه فیلم برداری کرده باشیم ، پس نرخ داده : $۷۸۶۴۳۲۰ (۵۱۲ * ۵۱۲ * ۳۰)$ بایت برثانیه و یا معادل بیش از ۶۰ مگابیت بر ثانیه است ، که آن خیلی بیشتر از مقادیر قابل ارسال با تکنولوژی های حال حاضر است .

۲- مشکل دوم از مختلف بودن الگوریتم های تحلیلی برای تصویر ایجاد می شود ، که آن نیز از گوناگونی شرایطی که برای تصویر پیش می آید ، از جمله : بارندگی ، مه ، سایه ، بازتاب ، تغییر روشنایی تصویر و نیز تفاوت شکل خودروها ، ناشی می شود . [20]

این دو مشکل با توجه به این که همه تصاویر نیاز به تحلیل ندارند ، کمرنگ می شوند . برای مثال اگر فرض کنیم که ماشینی سرعت ۱۰۰ کیلومتر در ساعت رادارد و طولش ۴ متر است ، اگر دوربین ما ۱۰۰ متر از خیابان را پوشش دهد و با سرعت ۱۰ فریم در ثانیه نیز فیلم بگیرد ما ۳۶ عکس از خودرو راداریم که برای آشکارسازی آن کافی می باشد . در ضمن ما نمی خواهیم کل تصویر را پردازش کنیم ، بلکه تنها بخش مورد نظر ROI (Region Of Interest) برای ما مهم است . [24]

۳- مشکل سوم و آخرفاکتور غیر اقتصادی بودن است. روش های پردازش تصویر نیازمند به : سخت افزار برای یک سیستم تصویری، یک دوربین ویدیویی مخصوص و یک کامپیوترسریع دارند، که در مقایسه با روش های مشهور و سنتی حلقه های کنترلی قیمت خیلی بالایی دارند و گران تمام می-شوند . باوجود این ، با توسعه کامپیوترها و ابزارهای ویدیویی و تصویری هزینه این عناصر سازنده سیستم به شدت کاهش پیدا کرده است .

دو نکته اصلی درمورد مطالعه در زمینه کنترل ترافیک مورد توجه است. اول کمک گرفتن از الگوریتم هایی که تنها قسمتی از تصویر را مورد استفاده قرار می دهند و دوم بکار بردن الگوریتم هایی که کل تصویر را مورد استفاده قرار می دهند. شایان ذکر است که ما روش اول را برای کم کردن مشکلات روش های پردازش تصویر را انتخاب کردیم . بر طبق این دو نظریه : هیچ سیستمی نیست که بتواند بر تمام ناملازمات غلبه کند ، از جمله :عوجاج های ناشی از فیلم برداری ، تغییر شرایط آب و هوایی، انعکاس ، سایه و روشنی تصویر ، اشکال مختلف ماشین ها و زمان پردازش و.... .

۵-۲- سیستم اندازه گیری میزان ترافیک

خلاصه روش :

در این روش ما برای هر خط ترافیک در تصویر، یک گروه از پیکسل ها را بر می داریم و آن ها را با حالت مرجع بدون ماشین مقایسه می کنیم . اگر مقدارشان تغییر کرده بود ، این امر را حاکی از مشاهدهء یک تصویر ماشین فرض می کنیم .

این الگوریتم شامل سه مرحله است :

۱- انتخاب پیکسل

۲- محاسبه مرجع ، بدون ماشین روی یک خط

۳- قانون آشکارسازی

خطوط یا همان ردیف های انتخابی برای حالت قبل از آمدن ماشین ها (حالت مرجع) به علت اینکه در طول روز میزان روشنایی تصویر تغییر می کند ، مقادیرشان مرتب باید تجدید شوند . ما برای

این کار یک سری تصویر از مواقع مختلف روز می گیریم و آن را به صورت آلبومی از مراجع در می آوریم ، سپس با توجه به زمان و ساعت ، تصویر i ام از آلبوم را به عنوان مرجع به برنامه معرفی می کنیم .

۵-۳- بیان مشروح روش

در ابتدا ما تصاویر ترافیکی را توسط یک دوربین فیلم برداری می گیریم و بعد ، از آن ها در بازه های زمانی که مدتش را در ادامه آرایه می دهیم ، عکس هایی را انتخاب می کنیم و پس از دیجیتال کردن ، آن ها را وارد کامپیوتر می کنیم . این عکس ها باید هم شامل عکس هایی باشد که ما آن ها را به عنوان آلبوم مرجع انتخاب می کنیم و هم شامل عکس هایی که ما از خطوط ترافیکی می گیریم. قابل ذکر است که ، تنها یک دفعه پر کردن آلبوم مرجع برای ما کافیست و از این پس ما تنها روی تصاویری که از ترافیک می گیریم ، اعمال بالا را انجام می دهیم .

حال که تصویرهایی هم به عنوان مرجع و هم به عنوان ترافیکی با خصوصیت های 256×256 پیکسل و بصورت سیاه و سفید با 256 سطح خاکستری در اختیار داریم ، بدین صورت عمل می کنیم : از آنجایی که می دانیم تصویر پس از دیجیتال شدن به صورت ماتریسی در می آید و هر عضو آن یک عدد است که مبین سطح خاکستری آن پیکسل از تصویر است ، برای شروع کار چند سطر از ماتریس مرجع را انتخاب کرده و در چند آرایه $L_1 \dots L_n$ می ریزیم . طرز انتخاب خطوط و نیز طرز انتخاب مقدار عددی n را در ادامه را شرح می دهیم.

حالا $L^1 \dots L^n$ را همان چند سطر ماتریس تصویر می گیریم (در همان محل ها) ، فقط با این تفاوت که این دفعه ماشین هم در تصویر داریم . پس مقدار سطح خاکستری L^i, L_i یعنی مقادیر اعضای آرایه هایشان متفاوت است . در اینجا نکته ای حائز اهمیت است و آن اینکه چون می دانیم ، بین این دو تصویر کمی نویز داریم ، مقدار آستانه ای (Threshold) را به نام U فرض می کنیم ، این بدین مفهوم است که ما دیگر اختلال نویزی با قدرت بیش از U نداریم .

سپس تک تک مقادیر دو آرایه را مقایسه (تفریق) می کنیم و حاصل را با مقدار U مقایسه می-کنیم ، در اینجا آرایه دیگری را بنام X با همان ابعاد آرایه های خطوط با مقدار پیش فرض صفر تعریف می کنیم . حالا اگر مقادیر تفاوت آرایه های خطوط بیش از مقدار U بود ، مقدار نظیر آن عضو را در آرایه X یک می کنیم و در غیر این صورت چون مقدار پیش فرض ما برای آرایه X صفر بود ، کاری نمی کنیم . اکنون که آرایه X مقداردهی شد و مقادیر اعضایش صفر و یا یک اند ، دست به مقایسه اعضای این آرایه می زنیم و هر عضو آرایه را با عضو کناریش مقایسه می کنیم . از این اعمال دو نتیجه بدست می آید :

۱- امکان تشخیص پهنای ماشین در هر خط با توجه به تعداد پیکسل هایی که در آرایه X یک شده اند، وجود دارد . در نتیجه می توان با آن نوع ماشین را تا حد زیادی تشخیص داد.

۲- پی بردن به اینکه چند دفعه مقدار آرایه X از صفر به یک و برعکس تغییر کرده است .

ما در اینجا با نتیجه دوم کار می کنیم و نتیجه اول می تواند در قسمت پیشنهادهایی که برای ادامه این روش مورد استفاده قرار می گیرد معرفی شود .

در ادامه کار با نتیجه دوم به اینجا رسیدیم که توانستیم تعداد لبه های ایجاد شده توسط حضور ماشین ها را با بکار بردن یک شمارنده برای یک خط حساب کنیم . اگر تعداد لبه های شمارش شده در یک خط را C بگیریم تعداد ماشین ها در این خط معادل $B=C/2$ است و اگر به همین صورت کار را برای خطوط دیگر نیز ادامه دهیم ، تعداد کل ماشین ها در تصویر دریافتی از رابطه زیر حاصل می شود :

$$B_i = 0.5 \times C_i \quad (1-5)$$

$$Total - numbers - of - car(s) = \sum_{i=1}^n B_i$$

در این فرمول B_i معرف تعداد ماشین در خط i ام است .

۴-۵- قانون محاسبه فاصله خطوط

فاصله خطوط آنقدر باید باشد که اولاً هر ماشین فقط در یک خط دیده شود و در ضمن حداقل در یک خط دیده شود (یعنی هر ماشین فقط و فقط یک خط را قطع کند و حداقل یک خط را قطع کند). برای این کار طول متوسط ماشین ها را L می گیریم و طبق این رابطه : $E_0 + L = L_{i+1} - L_i$ عمل می کنیم . بسته به مقدار E_0 و L فاصله خطوط را مشخص می کنیم .

۵-۵- خطای نسبی

این روش خطای نسبی از رابطه $Error = E_0/2L$ حاصل می شود . برای مثال اگر ما طول متوسط ماشین ها را ۴ متر و مقدار هرزی بین خطوط برای نیفتادن یک ماشین در دو خط متوالی (E_0) را ۵ متر بگیریم ، آنگاه فاصله خطوط معادل ۴٫۵ متر می شود . مقدار خطا با این مقادیر که بسیار نزدیک به مقادیر واقعی نیز هست ، معادل ۶٫۲۵٪ است ، که مقداری قابل قبول است . لازم به ذکر است ، مقدار خطا همان درصد شمرده نشدن یا دو بار شمرده شدن است .

۶-۵- نکات برتری این روش

۱- سادگی اجرای الگوریتم

۲- سرعت بالای اجرا (چون برنامه با حافظه کمی کار می کند .)

۳- وابسته نبودن شمارش به :

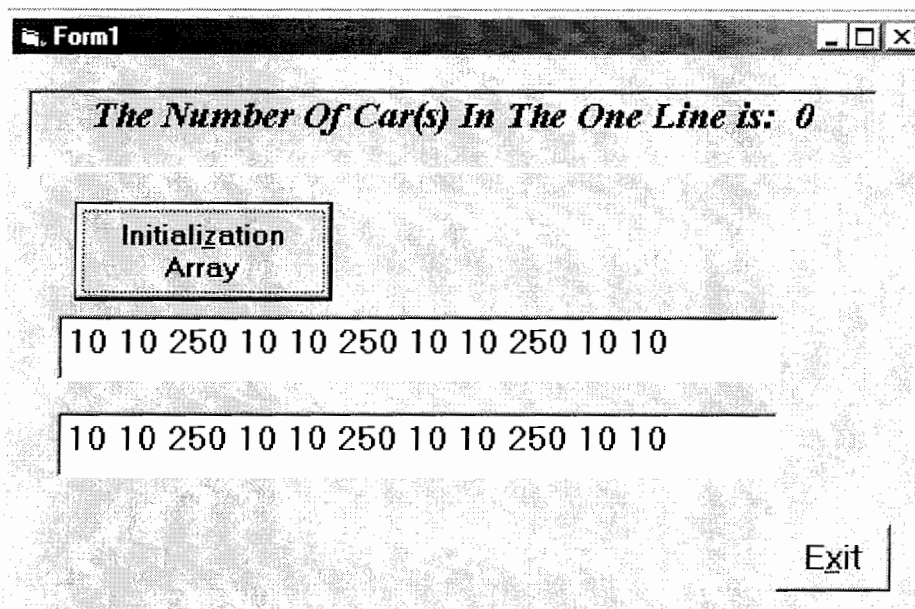
۳-۱- نوع و پهنای ماشین (چون ما لبه ها را می شماریم .)

۳-۲- طرز ایستادن از نظر : صافی و کجی و نیز قرارگیری بین خطوط سفید و یا خارج آن

و یا روی خطوط سفید .

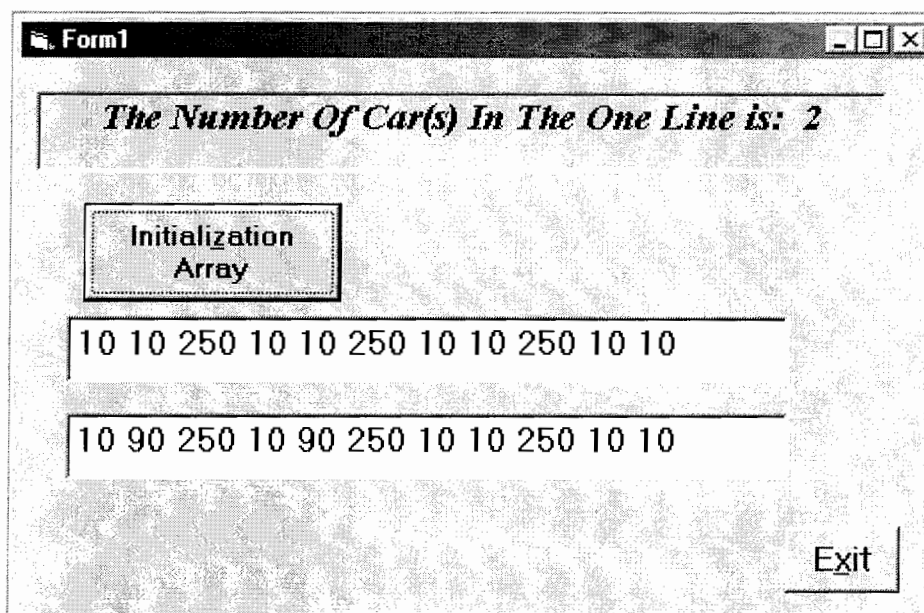
برای اثبات ادعایمان برنامه ای به زبان Basic و در محیط Visual Basic 6.0 نوشته ایم و تمام حالات ذکر شده در بالا را با این منطق مورد بررسی قرار داده ایم و خروجی ها را ارایه می-کنیم. کدهای برنامهء اثبات درستی این منطق در ضمیمه (۳) آورده شده است. [22]

اگر فرض کنیم ماشین ما دارای سطح خاکستری ۹۰ و خیابان مورد نظر ۴ بانده باشد و بجای چند پیکسل برای پهنای ماشین ما یک پیکسل را قرار دهیم که در جواب اثری ندارد و نیز خطوط سفید مقطع خیابان عدد ۲۵۰ را داشته باشند و آسفالت خیابان عدد ۱۰ و مقدار $U=20$ بگیریم، آنگاه به بررسی شرایط می پردازیم (البته فعلاً برای شمردن ماشین ها در یک خط اقدام می کنیم):



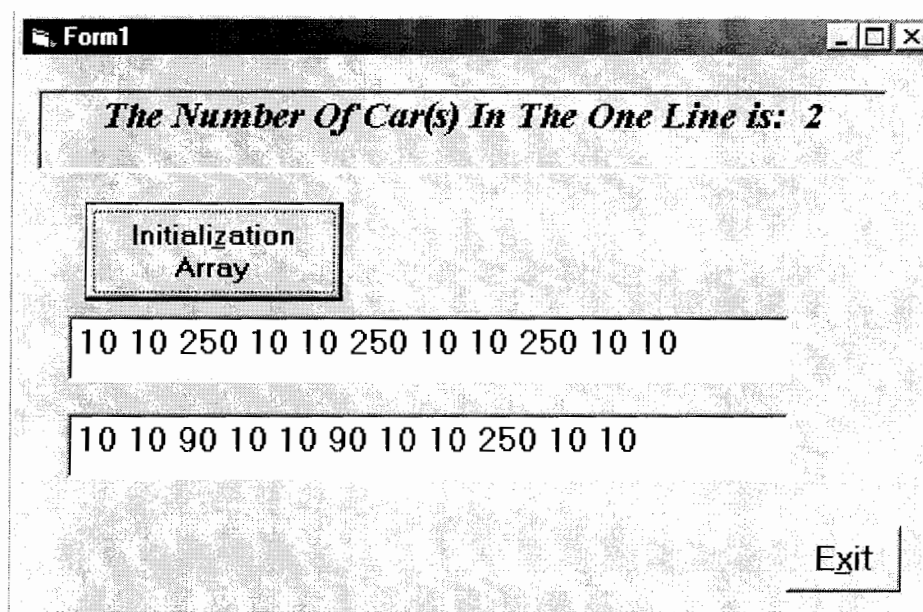
شکل ۵-۱ حالت اولیه برنامه اثبات درستی منطق جدید

حالت ۳- دو ماشین هر دو بین خطوط سفید باشند (چه صاف و چه مورب):



شکل ۴-۵ حالت ۳ برنامه اثبات درستی منطق جدید

حالت ۴- دو ماشین هر دو روی خطوط سفید باشند (چه صاف و چه مورب):



شکل ۵-۵ حالت ۴ برنامه اثبات درستی منطق جدید

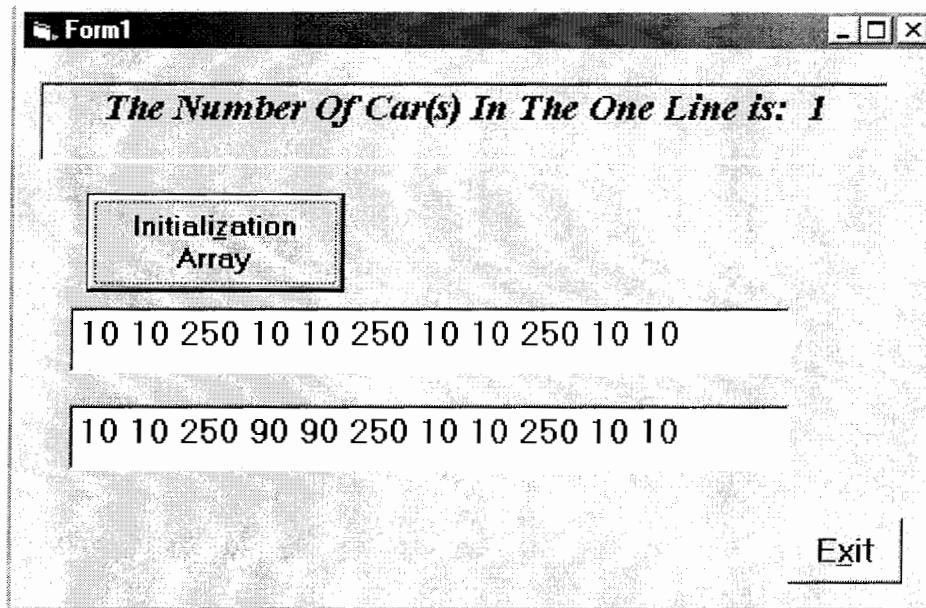
حالت ۵- دو ماشین یکی روی خطوط سفید و دیگری بین خطوط سفید باشد (چه صاف و چه مورب):

شکل ۵-۶ حالت ۵ برنامه اثبات درستی منطق جدید

حالت ۶- یک ماشین روی خطوط و دارای پهنای چند پیکسل باشد (چه صاف و چه مورب):

شکل ۵-۷ حالت ۶ برنامه اثبات درستی منطق جدید

حالت ۷- یک ماشین بین خطوط و دارای پهناى چند پیکسل باشد(چه صاف و چه مورب):



شکل ۵-۸ حالت ۷ برنامه اثبات درستی منطق جدید

نکته جالب آن که این منطق در تمامی حالات بالا به درستی جواب داده است و نکته دیگر آن که هیچ گاه دو ماشین کنار هم را یکی نمی شمارد، مگر آنکه کامل از پهلو به هم چسبیده باشند و کوچک بودن ماشین مخصوصاً از نظر پهنا بر کیفیت کار اثری نمی گذارد.

۵-۸- طرز پیاده سازی الگوریتم و نوشتن برنامه

همان طور که گفته شد ، برنامه ما نیاز به دریافت و خواندن تصویر ورودی و همچنین انجام عملیات -های تصویری و پردازشی دارد ، راه هایی که می شد برنامه را بوسیله آن نوشت به صورت زیر دسته بندی می شود :

- ۱- برنامه را با Visual Basic 6.0 کلاً بنویسیم ، ولی مشکل این کار دریافت شکل و نیز خواندن آن در آرایه بود ، که این برنامه در این زمینه انعطاف کمی داشت.
- ۲- برنامه را در محیط Matlab بنویسیم . بدون استفاده از توابع و زیر روال های Matlab ، که این کار طول و حجم برنامه را به طور وحشتناکی زیاد می کرد .
- ۳- برنامه را کلاً به زبان C بنویسیم آن هم در محیط مستقل C ، که این روش هم تنها کمی بهتر از روش دوم بود .

- ۴- برنامه را در محیط Matlab بنویسیم . و نیز با استفاده از توابع و زیر روال های Matlab ، که این کار از همه معقول تر و به صرفه تر بود و طول و حجم برنامه را به طور غیر قابل باوری کم می کرد و ما همین روش را انتخاب کردیم . [21]

۵-۹- نتیجه بحث

ما برای اثبات درستی ادعایمان در مورد این الگوریتم جدید نیاز به حل اولیه چند مثال در حالات مختلف داشتیم ، تا خواص و مزایای این روش را شرح داده و اثبات کنیم ، برای نیل به این منظور ابتدا همان طوری که مشاهده گردید ، برنامه ای را به صورت خلاصه برای شمردن ماشین ها در یک خط نمونه ، با این الگوریتم و روش خاص به زبان Basic نوشتیم . تا اینکه تمام حالاتی که ممکن است برای یک تصویر ترافیکی پیش بیاید را بررسی کرده باشیم . خروجی برنامه به صورت Visual حاکی از صحت روش حتی در حالات خاص و نادر بود .

سپس برای کلیت بخشیدن به کار و نیز دریافت و پردازش تصویرهای واقعی برنامه ای را مجدداً در محیط نرم افزاری Matlab نوشتیم و باز هم ابتدا از شمردن ماشین ها در یک خط شروع کردیم و

پس از دریافت صحیح تعداد آنها در یک خط این برنامه را به صورت یک تابع (Function) شمارنده ماشین در یک خط در آورده و سپس با صدا زدن (Calling) این برنامه به دفعات متعدد تعداد کل ماشین ها در تصویر دریافتی را محاسبه کردیم . یعنی بعد از جمع زدن تمامی مقادیر خروجی های تابع مورد نظر مقدار کل ماشین ها یا همان Total Numbers Of Car(s) را در یک تصویر حساب کردیم .

قابل ذکر است ، کدهای برنامه شمارش ماشین ها در یک خط در محیط Matlab در ضمیمه (۴) ، به همراه یک نمونه مثال با خروجی های مطلوب آورده شده است و نیز کل برنامهء شمارش ماشین های درون تصویر در محیط Matlab در ضمیمه (۵) آورده شده است . [23]

۵-۱۰- روش تعیین تعداد خطوط

روش تعیین تعداد خطوطی از تصویر که باید به عنوان نمونه انتخاب شوند ($L1..Ln$) تا در آرایه های قرار گیرند و بعد از آمدن ماشین مورد قیاس واقع شوند ، بدین صورت است :

در این قسمت ما سعی می کنیم تا با یک مثال روش تعیین تعداد خطوط را بطور کامل و واضح شرح دهیم .

فرض کنیم که ما طول متوسط ماشین ها را $L=4m$ گرفته ایم و تصویر ما $P=90m$ از طول جاده را پوشش دهد، چون می دانیم ، تصویر دیجیتال ما $256*256*8$ است (یعنی دارای ابعاد $256*256$ پیکسل بوده و نیز 256 سطح خاکستری دارد.)، پس طول 90 متری جاده معادل 256 سطر ماتریس است و از آنجایی که ما مقدار $E0=0.5m$ می گیریم ، پس طبق رابطه زیر :

$$E0+L=Li+1-Li \quad (2-5)$$

فاصله خطوط متوالی معادل $4,5$ متر می شود و حداکثر به مقدار $P/(Li+1-Li)=90/4.5=20$ سطر ماشین در تصویر داریم ، در نتیجه باید 20 خط ($L1..L20$) را برای شمردن انتخاب کنیم . این بدان مفهوم است که ما باید هر $256/20$ یا معادل هر 13 سطر ، یک سطر نمونه گیری کنیم و در Li ذخیره کنیم . به همین خاطر ما کل تصویر را با یک حلقه تکرار و تابع اصلی که تعداد ماشین ها را در

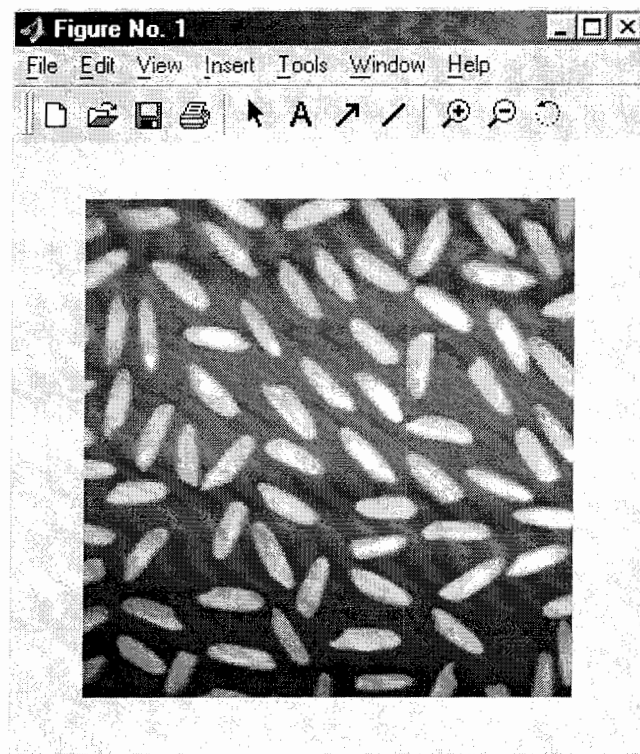
یک خط می شمارد ، با $Step=13$ از ۰ تا ۲۵۵ صدا می زنیم و در نهایت در شمارنده (Counter) ، Totalnumcars ما تعداد تمام ماشین ها را در تصویر ارایه می دهیم .

ممکن است به نظر آید که ما باید برای هر تصویر بدست آمده هر دفعه یک سری تنظیماتی در برنامه انجام دهیم ، ولی این درست نیست و یک دفعه تنظیم کافیسیت ، چون محل استقرار دوربین ها و متعاقباً میدان دید آن ها (P) ثابت است و نیز ما خودمان همیشه ابعاد تصویر دیجیتال را $۲۵۶*۲۵۶$ می گیریم . با توجه به تمام این خصوصیات برای هر دوربین تنها یک بار تنظیم مقادیر برنامه کافیسیت .

۵-۱۱- مزایا و کاربردهای روش

غیر از مزایایی که در صفحات قبل برای این روش ذکر شد ، مزایای دیگری نیز برای آن متصور است از جمله اینکه : بر خلاف دیگر الگوریتم هایی که تا بحال ارایه شده است و ما از آن ها در فصل قبل یاد کردیم ، ما از پارامترهایی نظیر سرعت ، ابعاد ، طرز حرکت و حتی ظاهر ماشین ها استفاده نکرده ایم و این خود دلیلی بر سوق دادن ما به این سمت می شود که از این روش برای شمردن به عنوان مثال : تعداد عابرین در پیاده رو و یا شمردن هر چیز دیگری در تصویر استفاده کنیم .

به عنوان مثال شمردن تعداد دانه های برنج در تصویر صفحه بعد که با وجود بی نظمی زیاد آن ها این روش کارایی خودش را با دادن جوابی قابل قبول ، نشان داده است . جوابی که از این روش برای شمارش تعداد دانه های برنج در تصویر صفحه بعد حاصل شد ، عدد ۸۹ بود این عدد با مقدار واقعی که چیزی حدود ۷۹ عدد برنج بود ، تفاوتی نمی کند و با توجه به تمام موارد بالا ، جوابی قابل قبول با ۱۱٫۲٪ خطا است .



شکل ۵-۹ نمونه ای از کاربردهای متنوع این روش در شمارش تعداد دانه های برنج را نمایش می دهد .

نباید فراموش کرد که مهم ترین کاربرد این روش ، کنترل ترافیک حاصل از تجمع خودروهاست . بدین معنی که باعث ایجاد تعادل و پخش ترافیک خیابان ها و جلوگیری از تجمع خودروها در یک محل و خالی ماندن محل های دیگر می شود .

۵-۱۱-۱- مزایای دیگر

از دیگر مزایای این روش ، کم بودن حجم برنامه و نیز کوتاهی زمان اجرای آن است ، این امر به ما برای انجام همزمان دریافت تصاویر و نیز پردازش آن و ارایه خروجی و اعمال آن بر کنترلرهای موجود به صورت سلسله وار کمک می کند .

۵-۱۲- سیستم ترافیک

در کنترل همزمان ترافیک نکته ای نهفته است و آن لختی سیستم است ، یعنی لزومی ندارد که تمامی فریم های تصویر دریافتی را مورد پردازش قرار دهیم ، چون ترافیک یک خیابان هیچ گاه به سرعت تغییر چهره نمی دهد و یک دفعه کم و زیاد نمی شود . این بازه زمانی حداقل ۵-۱۰ دقیقه طول می کشد تا شکل ترافیک برای اعمال کنترل ، عوض شود و این امر به ما کمک می کند که :

- ۱- تنها چند فریم از هر ساعت فیلم تصویر ترافیکی برای ما کافی باشد .
- ۲- این فرصت را به ما می دهد تا برنامه را اجرا و داده های خروجی آن را به کنترلرهای موجود اعمال کنیم و ما را در انجام عملیات همزمان یاری می کند .

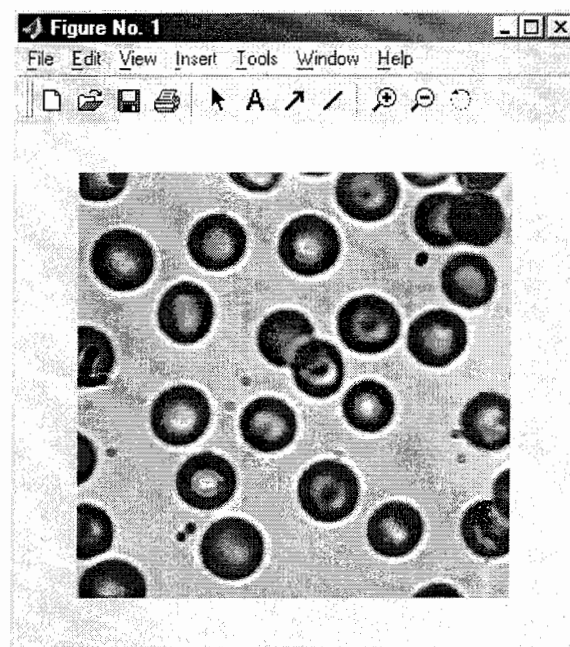
۵-۱۳- طرز استفاده از نتایج حاصله

حال ما با بدست آوردن تعداد خودروها در یک تصویر از خیابان می توانیم با توجه به دیگر تصاویری که از خیابان های جانبی و منتهی به این خیابان بدست می آوریم ، چراغ های سبز و قرمز چهارراه را از نظر مدت زمان روشن و خاموش بودن در برابر هر خیابان کنترل کنیم و بطور مثال ، اگر یک خیابان دارای حجم ترافیکی زیاد (تعداد خودروی بیشتر) است ، به همان نسبت بیشتر بودن تعداد خودروها ، زمان سبز بودن چراغ راهنمایی را بیشتر کنیم .

۵-۱۴- مثال دیگر

در اینجا مثالی دیگر از کاربردهای این روش ، در زمینه شمارش تعداد گلبول های قرمز خون در یک تصویر را در نظر گرفته ایم که تنظیمات برنامه و همچنین مقدار خروجی در ضمیمه (۵) آمده است .

شکل مورد نظر در این مثال دارای حدوداً ۳۳ گلبول قرمز است و جواب این روش ۳۷,۵ شد . شاید به نظر آید که چرا این عدد اعشاری شد؟ ، این بدان علت است که در گوشه های تصویر گلبولی وجود دارد که باعث ایجاد تنها یک لبه شده است و آن لبه دیگر ، پایین نیامده است و هنگام تقسیم کردن لبه ها بر دو ، عدد ۰,۵ حاصل شده است . با توجه به کیفیت کار این روش و خطای ۱۲٪ در این شکل نامنظم ، این مطلب از اهمیت کمی برخوردار است .



شکل ۵-۱۰ نمونه دیگری از کاربردهای متنوع این روش در شمارش تعداد گلبول های قرمز خون را نمایش می دهد .

۵-۱۵- حل مثالهای اولیه با تصاویر واقعی از بزرگراه ها با این روش

حال ما در این قسمت از پایان نامه بعد از این که سه مرحله زیر را انجام دادیم یعنی :

۱- شبیه سازی الگوریتم جدید به زبان Visual Basic در هفت حالت ممکنه و در یک صفِ خودرو در بزرگراه.

۲- شبیه سازی و بکاربردن الگوریتم برای کاربردهای دیگر و تشخیص تعداد اشیاء دیگری غیر از خودرو در تصویر و اعلام نتایج آن .

۳- تعیین میزان خطای این روش در این سری از تصاویر.

اکنون می خواهیم الگوریتم را روی تصاویری که البته در این مرحله بطور کامل عمود نیستند و از یک بزرگراه دریافت شده اند امتحان کنیم ، تا ببینیم : اولاً نقایص احتمالی کارمان در چه مواردی است و ثانیاً متوجه شویم که آیا در صد خطای ما برای تصاویری نظیر خودروی در حال حرکت هم هنوز همان مقدار حدود ۱۰٪ است ، یا نه ؟ تا با توجه به این دو مورد بتوانیم در حد امکان این نقایص را جبران کرده و در جهت ارتقاء کیفیت این روش بکوشیم .

البته همان طور که گفته شد ، در این چند مثال اولیه ، هنوز شرایط ۱۰۰٪ همان شرایط مطلوب و مورد نظر این الگوریتم نیست ، چرا که پایه های این روش بر روی یک سری فرضیات استوار است و مسلم است که اگر شرایط و قید های مورد نظر آن محقق نشود ، جواب ما به یقین دارای خطای بیشتری از حالات تئوری و مفروض خواهد شد و مقدار خطاها از مقدار خطاهای شمارش ، در قسمت کاربردهای دیگر که حدود ۱۰٪ بود ، بیشتر شود .

از جمله موارد رعایت نشده در این سری مثالهای اولیه را می توان :

۱- شرط کاملاً عمود بودن دوربین دیجیتال بر سطح جاده .

۲- پوشش کامل جاده بدون لبه ها و قسمت فضای سبز و جداول کناری جاده .

ذکر کرد .

اگر این موارد به همراه فرضیات دیگر و یک سری محاسبات جاده کاملاً رعایت می شد، بدون شک خطاهایی از جمله: خطا بر اثر اینکه یک خودرو تا نیمه در تصویر باشد (از پهلو) و شمرده نشود، پدیدار نمی گشت. این امر به خودی خود میزان خطا را تا حد زیادی در این تصاویر بالا برده است.

حال که بحث اولیه را در مورد این مثالها انجام دادیم، بهتر است به بررسی طرز تسخیر این تصاویر و نکات دیگری بپردازیم و در مرحله نهایی (سوم) نیز روی هر شکل جداگانه بحث کرده و تعداد واقعی و اندازه گیری شده با این روش را نشان دهیم. همچنین قصد داریم، متوسط درصد خطای نسبی و در صد کل خطا را با روش حداقل مربعات خطا (MSE) ارائه دهیم. تا بتوانیم آن را با مقادیر بدست آمده از قسمتهای قبل و نیز قسمت شبیه سازی (تئوری) مقایسه کنیم.

۱۶-۵- طرز تسخیر تصاویر

ما برای تسخیر و آماده سازی تصاویر مورد بحث با این مثالها مراحل زیر را انجام دادیم:

۱- ابتدا توسط دوربین دیجیتال دانشگاه تصاویری را بمدت ۱۵-۲۰ دقیقه از بزرگراه مقابل دانشگاه با رفتن بر روی پل عابر پیاده به صورت تقریباً عمود (۶۰ تا ۷۰ درجه) گرفتیم و از آنجایی که دوربین مورد بحث با کیفیت بسیار بالا و سرعت بالا کار می کرد، تصاویر دریافتی کاملاً مناسب بودند، و حرکت خودرو در تصویر باعث بلوری شدن عکس نشده و نیاز ما را به پردازش اولیه تصویر بر طرف می کرد. تصویر برداری برای اجتناب از لرزش دست، بر روی پایه انجام شد و هر چند دقیقه یک بار هم از آنجا که دوربین دارای 8MB حافظه جداگانه بود، یک عکس از جاده گرفتیم. این عکسها هم شامل تصاویری از ساختار جاده، که خالی از وسیله نقلیه است و هم تصاویری از جاده با حضور خودروها بودند. این عمل را در چند محل از بزرگراه انجام داده و چند سری تصویر بدست آمد.

۲- در این مرحله خوشبختانه چون این کارت حافظه 8MB قابل اتصال مستقیم به کامپیوتر بود، اطلاعات (عکسها) را روی یک CD ریخته و به منزل آوردم.

۳- در این مرحله ، تصویرها آن تصویرهایی نبودند که برنامه می توانست به عنوان ورودی خودش قبول کند ، چون دارای مشخصاتی غیر از مشخصات مفروض برنامه بودند . از آن جمله :

۱-۳- تصاویر دارای ابعاد بزرگ بودند . مثلاً : 1030×1320 .

لازم به ذکر است ، برای برنامه ما مهم نیست که ابعاد تصویر چند باشد ، ولی از آنجایی که باید برای تصاویر ورودی یک اصول و استاندارد را رعایت کنیم ، تا بعدها بشود از این برنامه بطور اتوماتیک استفاده کرد ، پس به ناچار نیاز به حفظ یک ابعاد مشخص هستیم . از این مورد نمی توان به عنوان اشکال برنامه یاد کرد ، چون اگر ما لنز و زاویه و نیز خود دوربین را با منوهایش تنظیم کنیم ، (تنظیم Resolution) ، دیگر نیازی به تغییر اندازه تصویر (Resize) نداریم .

ولی حالا که ما تصاویری با این مشخصات داریم آنها را با نرم افزار Acd see به اندازه دلخواه در می آوریم و ابعاد آنها را روی ۲۵۶ عرض و ۲۵۶ یا ۱۹۲ یا ۱۴۵ ارتفاع قرار می دهیم یعنی نسبت عرض به ارتفاع : ۱ یا ۱,۳۳ یا ۱,۷۷ . ولی هنوز تصویر ما مشکل دارد که مرحله دوم را انجام می-دهیم .

۲-۳- ما فرض کرده بودیم که تصاویر سیاه و سفید و با ۸ بیت سطح خاکستری (معادل ۲۵۶ سطح خاکستری) باشند . اما تصاویر دریافتی ما رنگی با رنگ اصلی (True color) بودند . برای حل این مشکل هم از نرم افزار Acd see کمک گرفتیم .

بعد از انجام این سه مرحله برای دریافت تصاویر، هر تصویر ما دارای مشخصات زیر شد:

۱- ۸ بیت سطح خاکستری (معادل ۲۵۶ سطح خاکستری)

۲- ابعاد مناسب 256×256 یا با ارتفاع کمتر ۱۹۲ یا ۱۴۵ پیکسل

۳- پسوند تمام عکسها jpg بود که نیازی هم به تبدیل به پسوندهایی نظیر tif یا غیره را نداشت . چون کیفیت عکسها را پائین می آورد و نیز توان تحلیل بادقت مناسب را از برنامه می-گرفت .

سپس ما تصویرهایی داشتیم که در سه موقعیت گرفته شده بودند و هر موقعیت (حالت) ، هم شامل تصاویر بدون خودرو بود و هم شامل تصاویر با خودرو .

حالت ۱) عکس به نام فایل emptyA.jpg به عنوان مرجع بدون ماشین در نظر گرفته شد ، که بیانگر موقعیت (حالت) اول دوربین بود . سپس سه عدد عکس به نام فایل‌های fullA.jpg و fullB.jpg و fullC.jpg که هر یک بیانگر یک عکس با خودرو مربوط به همان موقعیت بودند ، دریافت شد .

ما این سه عکس را جداگانه با عکس مرجع مورد مقایسه قرار دادیم و با برنامه ، خروجی آنها را بدست آوردیم . لازم به ذکر است محاسبات مربوط به طول گام و نیز مقدار حد آستانه طبق تئوری گفته شده در قبل انجام شده است . متن برنامه بطور نمونه به همراه خروجی ، برای یک عکس در ضمیمه ۶ آمده است .

خروجی های بدست آمده مطابق جدول زیر هستند :

جدول ۵-۱ موقعیت اول ونتایج حاصل از تحلیل برنامه روی سه عکس

نام فایل مرجع	نام فایل تصویر مورد تحلیل	تعداد واقعی	تعداد محاسبه شده	در صد خطای نسبی هر تصویر
emptyA.jpg	fullA.jpg	2	1	50%
emptyA.jpg	fullB.jpg	1	1	0%
emptyA.jpg	FullC.jpg	1	1	0%

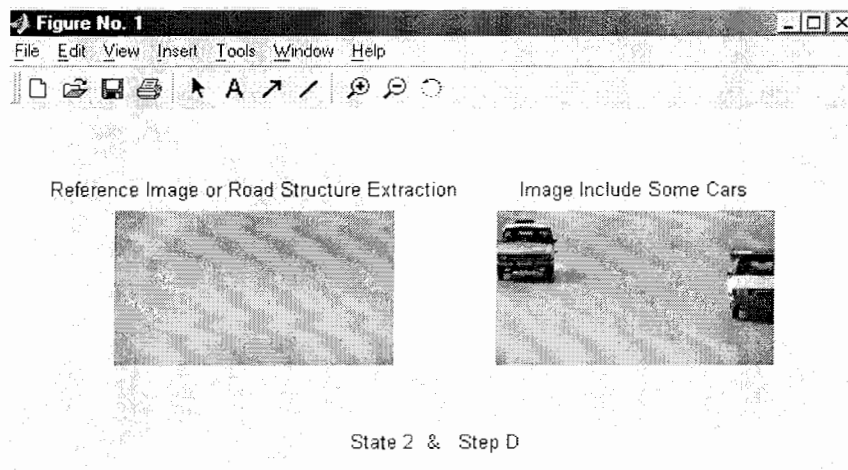
برای مشاهده تصاویر واقعی نیز در شکل‌های بعدی به ترتیب ، تصاویر هر دو زوج فایل را برای هر بار تحلیل آورده ایم :

حالت ۲) عکس به نام فایل emptyD.jpg به عنوان مرجع بدون ماشین در نظر گرفته شد ، که بیانگر موقعیت (حالت) دوم دوربین بود . سپس یک عدد عکس به نام فایل fullD.jpg که بیانگر یک عکس با خودرو مربوط به همان موقعیت بود ، دریافت شد . بعد از انجام مراحل مشابه مراحل قبل ، جدول زیر برای موقعیت دوم بدست آمد :

جدول ۵-۲ موقعیت دوم ونتایج حاصل از تحلیل برنامه روی یک عکس

در صد خطای نسبی در این تصویر	تعداد محاسبه شده	تعداد واقعی	نام فایل تصویر مورد تحلیل	نام فایل مرجع
50%	1	2	fullD.jpg	emptyD.jpg

و تصویر واقعی مربوط به موقعیت (حالت) دوم و مرحله D به صورت زیر است :



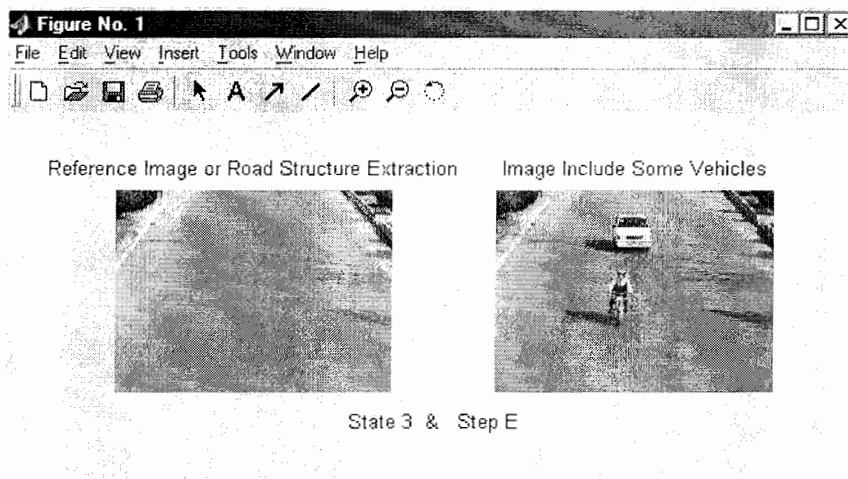
شکل ۵-۱۴ موقعیت (حالت) دوم - تصویر (مرحله D)

حالت ۳) عکس به نام فایل emptyE.jpg به عنوان مرجع بدون ماشین در نظر گرفته شد ، که بیانگر موقعیت (حالت) سوم دوربین بود . سپس یک عدد عکس به نام فایل fullE.jpg که بیانگر یک عکس با خودرو مربوط به همان موقعیت بود ، دریافت شد . بعد از انجام مراحل مشابه مراحل قبل ، جدول زیر برای موقعیت سوم بدست آمد :

جدول ۳-۵ موقعیت سوم ونتایج حاصل از تحلیل برنامه روی یک عکس

نام فایل مرجع	نام فایل تصویر مورد تحلیل	تعداد واقعی	تعداد محاسبه شده	در صد خطای نسبی در این تصویر
emptyE.jpg	fullE.jpg	2	2	0%

و تصویر واقعی مربوط به موقعیت (حالت) سوم و مرحله E به صورت زیر است :



شکل ۵-۱۵ موقعیت (حالت) سوم - تصویر (مرحله E)

۵-۱۷- نتایج

از کل این بحث و مثالها به یک سری نتیجه گیری و جمع بندی به شرح زیر می رسیم :

۱- با وجود اینکه تصاویر به دلایلی که در صفحات قبل ذکر شد ، مطابق با شرایط و فرضیات برنامه نبود ؛ متوسط خطای نسبی ۲۰٪ می باشد .

توجه : این مقدار خطا ، اگر تعداد خودروها بیشتر بود حتی با همین شرایط نامساعد فیلم برداری ، مقدار کمتری پیدا می کرد .

۲- محاسبه خطا به روش حداقل مربعات خطا (MSE) به ما مقدار ۱۴,۴٪ را می دهد .

۳- دلیل ایجاد خطای ۵۰٪ در مقدار خروجی برنامه در تصاویر A,D بدلیل وجود خودروها به صورت نیمه (از پهلو) در تصویر است ، که اگر ما آن شرط پوشش صحیح کل جاده را رعایت می کردیم ، دیگر خودرویی نصفه در تصویر یافت نمی شد . که این امر خود باعث خواهد شد ، این نوع خطاها از بین بروند .

۵-۱۸- حل مثالهای نهایی با تصاویر واقعی از بزرگراه ها با این روش

در این قسمت از پایان نامه بعد از این که ما یک سری مثالهای اولیه را در مورد صحت کار این الگوریتم در قسمت (۵-۱۵) مورد بررسی قرار دادیم، و نشان دادیم که اگر شرایط کاری برای این الگوریتم محیا نباشد، جوابهای حاصل از آن زیاد قابل اعتماد نیست، می خواهیم در این قسمت بعد از اعمال شرایط مناسب بر تصاویر ورودی نتیجه را مورد بررسی قرار داده و با حالات قبل و نیز با مقدار خطا در قسمت تئوری (کمتر از ۱۰٪) مقایسه کنیم تا رأی نهایی در مورد مناسب بودن و یا عدم مناسب بودن این الگوریتم، برای این امر خاص و هدف مشخص را صادر نماییم.

لذا برای نیل به این هدف بعد از مراجعات مکرر به شرکت کنترل ترافیک تهران (TTCC) توانستیم حدود ۱,۵ ساعت کلیه دوربین های آن ها را در اختیار گرفته و روی یک سری از دوربینها که شرایط کاری برنامه را برآورده می کرد و مناسب کار ما بود رفته و فیلم بگیریم. دوربینهای این شرکت از آنجایی که سه خاصیت مهم داشت برای کار ما مناسب تر از حالتی بود که خودمان فیلم بگیریم : اولاً: ارتفاع زیاد دوربین به علت نصب آن بر روی دکل های بلند و ثانیاً: عدم حرکت در صورت نیاز و قدرت مانور روی بزرگراه های اطراف برای انتخاب تصویر بهتر و در نهایت توانایی موازی شدن لنز این دوربینها با سطح جاده.

بعد از گرفتن این فیلم مشاهده کردیم که فیلم از کیفیت بالایی برخوردار نیست، ولی از آنجایی که کار ما مقایسه دو تصویر است، لذا کیفیت هر دو تصویر پایین خواهد بود و این بر نتایج ما اثری نخواهد گذاشت .

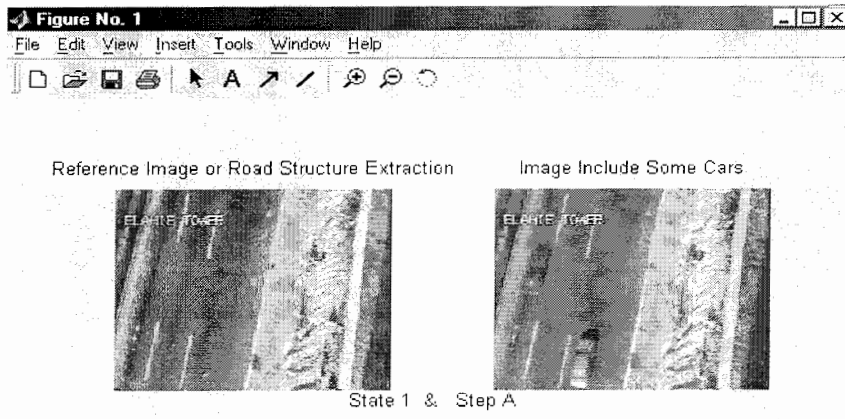
باز هم ما همان مراحل کاری در قسمتهای ۵-۱۵ و ۵-۱۶ را انجام داده و یک سری تصویر به عنوان مرجع و یک سری تصویر به عنوان تصاویر مورد تحلیل بدست آوردیم. در این فیلم ما ۶ مکان را مورد تحلیل قرار دادیم، پس ۶ عکس مرجع داریم و در کل ۲۲ بار برنامه را بر روی آنها اجرا کردیم، در واقع ۲۲ مثال با شرایط مطلوب برنامه حل نمودیم که در ادامه نتایج را خواهیم دید. ما تک تک مثال ها را ارائه خواهیم کرد و در صورت نیاز به توضیح، آنها را توضیح می دهیم.

روی بزرگراه اول یعنی موقعیت A برج الهیه ما ده بار برنامه را اجرا کردیم که نتایج آن در جدول ۴-۵ آمده است :

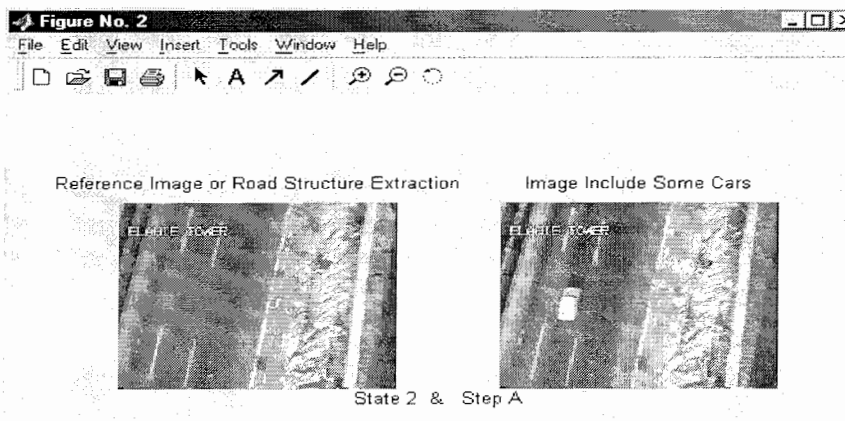
جدول ۴-۵ موقعیت اول (A برج الهیه) و نتایج حاصل از تحلیل برنامه روی ده عکس

در صد خطای نسبی هر تصویر	تعداد محاسبه شده	تعداد واقعی	نام فایل تصویر مورد تحلیل	نام فایل مرجع
0%	2	2	FullelahA1.JPG	emptyelahA.JPG
0%	1	1	FullelahA2.JPG	emptyelahA.JPG
0%	3	3	FullelahA3.JPG	emptyelahA.JPG
20%	5	4	FullelahA4.JPG	emptyelahA.JPG
14.2%	7	6	FullelahA5.JPG	emptyelahA.JPG
25%	4	3	FullelahA6.JPG	emptyelahA.JPG
0%	2	2	FullelahA7.JPG	emptyelahA.JPG
50%	2	1	FullelahA8.JPG	emptyelahA.JPG
20%	5	4	FullelahA9.JPG	emptyelahA.JPG
0%	6	6	FullelahA91.JPG	emptyelahA.JPG

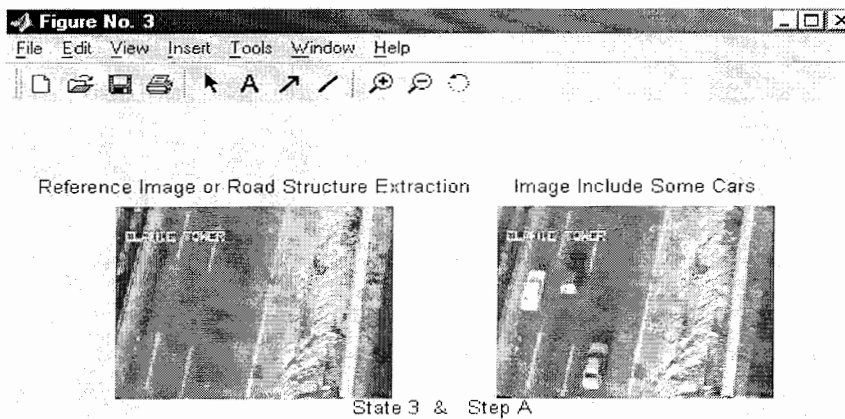
برای مشاهده تصاویر واقعی نیز در شکل‌های بعدی به ترتیب ، تصاویر هر دو زوج فایل را برای هر بار تحلیل آورده ایم :



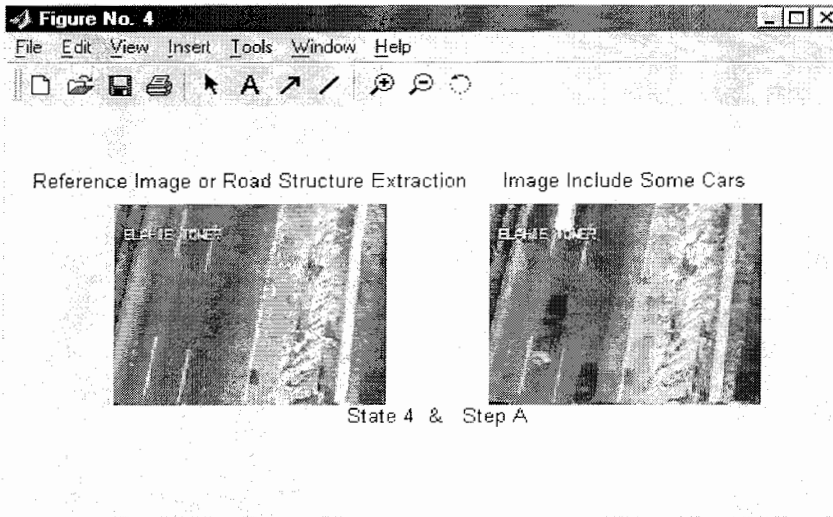
شکل ۵-۱۶ موقعیت اول - تصویر (مرحله ۱)



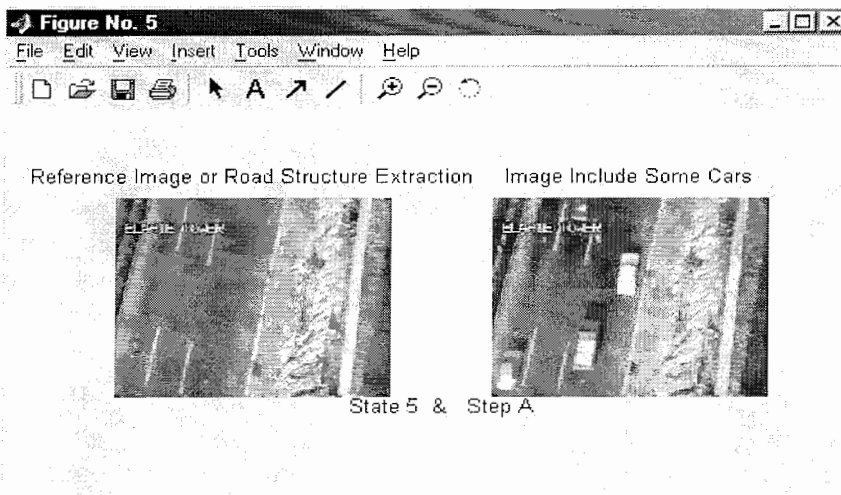
شکل ۵-۱۷ موقعیت اول - تصویر (مرحله ۲)



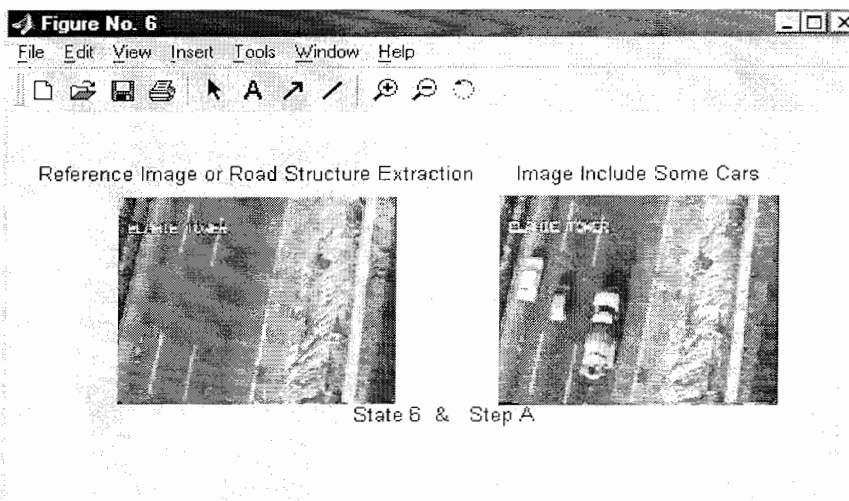
شکل ۵-۱۸ موقعیت اول - تصویر (مرحله ۳)



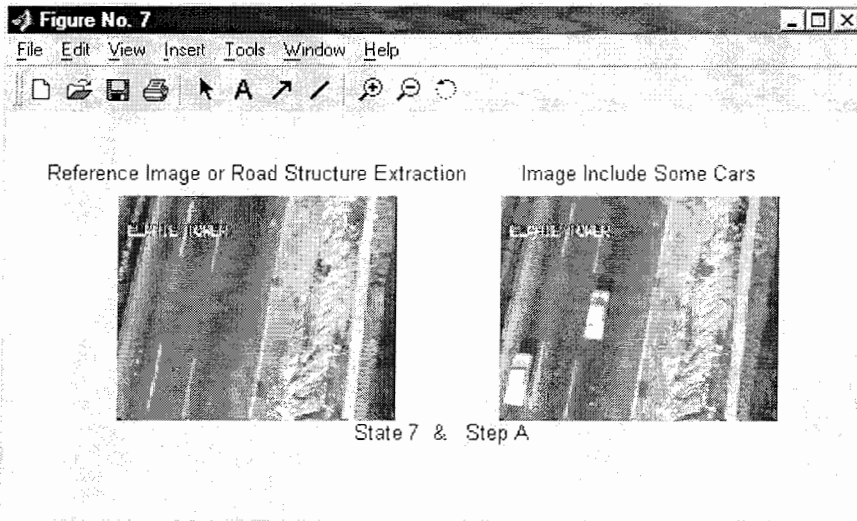
شکل ۵-۱۹ موقعیت اول - تصویر (مرحله ۴)



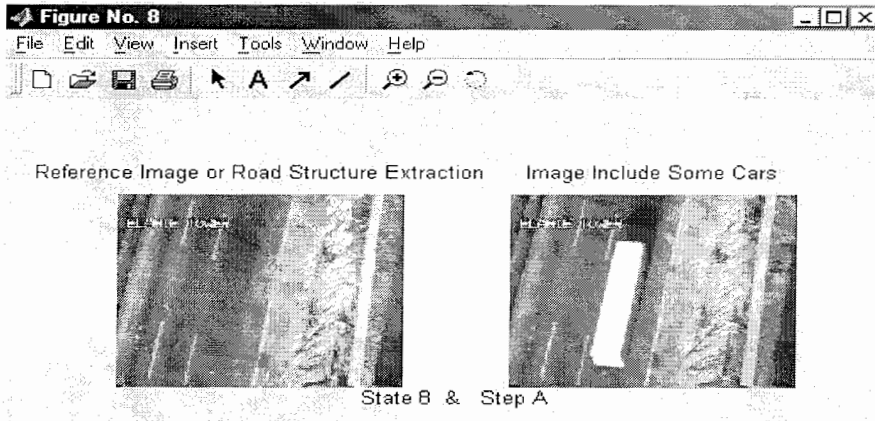
شکل ۵-۲۰ موقعیت اول - تصویر (مرحله ۵)



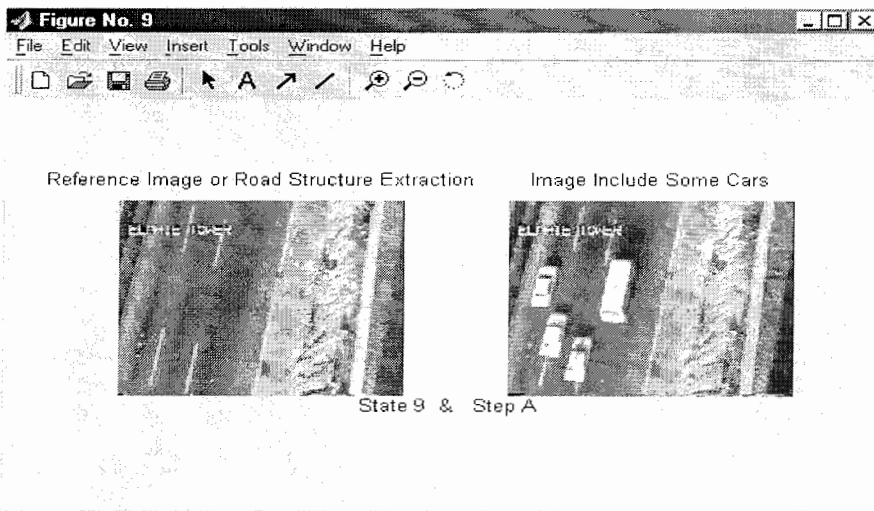
شکل ۵-۲۱ موقعیت اول - تصویر (مرحله ۶)



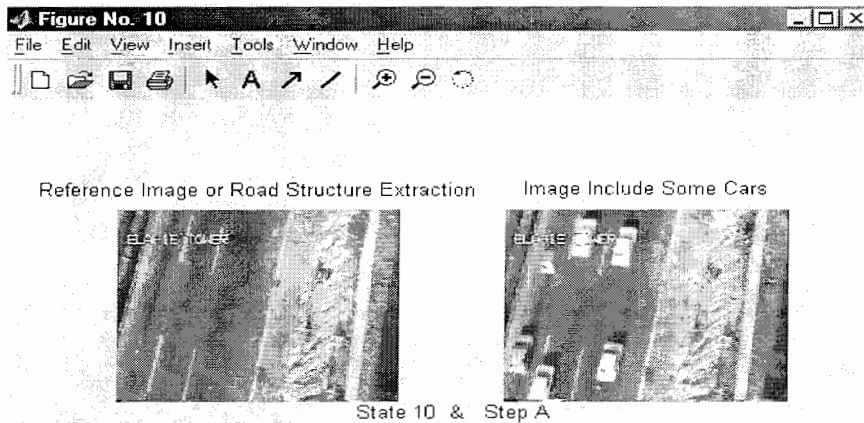
شکل ۵-۲۲ موقعیت اول - تصویر (مرحله ۷)



شکل ۵-۲۳ موقعیت اول - تصویر (مرحله ۸)



شکل ۵-۲۴ موقعیت اول - تصویر (مرحله ۹)



شکل ۵-۲۵ موقعیت اول - تصویر (مرحله) ۱۰

تحلیلها و نتایج حاصله به دو بخش تقسیم می شود: ۱- تحلیل هایی بدون خطا ۲- تحلیل هایی با خطا. مسلم است که برای حالات بدون خطا نیازی به توضیح نیست ولی برای حالات با خطا شرح دادن علت وقوع خطا خالی از لطف نیست:

خروجی برنامه در حالات: ۴ و ۵ و ۶ و ۸ و ۹ مقداری خطا دارد، که آنها را تشریح می کنیم:

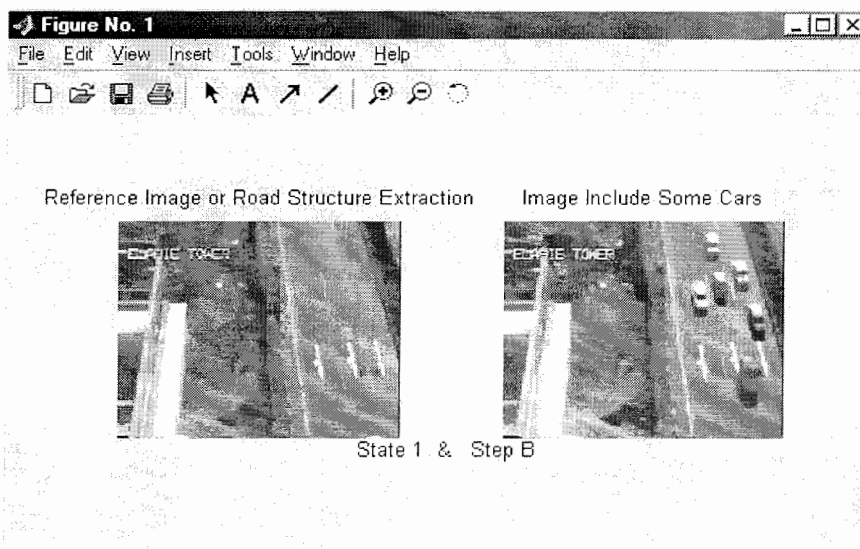
خطای برنامه در تمام حالات به علت وجود نوعی کشیدگی و یا بعبارتی عدم تناسب مابین خوددوری شمارش نشده و یا دو بار شمارش شده با سایر خودروها در تصویر دارد. به عنوان مثال این امر در تصویر ۶ چون کامیون داریم، مشهود است. این خودرو در حین شمارش دوبار شمرده شده که خود باعث خطا در نتیجه تحلیل است و نمونه بارزتر آن در تصویر ۸ مشاهده می شود که اتوبوسی، دو بار باز هم به همان علت، شمرده شده است. شاید به نظر آید که ما در این طور تصاویر همیشه مقداری خطای شمارش داریم، ولی اگر ما به هدف غایی از این تحلیلها بنگریم دیگر این طور سئوالات پیش نخواهد آمد، زیرا هدف نهایی ما از این تحلیلها بیان میزان ازدحام در بزرگراههای موجود است، و همانطور که بدیهی است ماشینی که دو یا سه بار حین تحلیل به علت کشیدگی شمرده شود، مسلماً به اندازه دو یا سه خودروی سواری جا می گیرد و ازدحام ایجاد می کند و در واقع از آنجایی که ما نهایتاً تعدادی که در خروجی برنامه ارائه می دهیم، همان تعداد معادل خودروی سواری است، برنامه برای بیان حالت و میزان ازدحام درست کار می کند.

اکنون که ما برای یک موقعیت تمام حالات ممکن را آوردیم و برای حالاتی که در خروجی برنامه خطا پیش می آید، توضیح لازم را دادیم، موقعیتهای دیگری را مورد بررسی قرار خواهیم داد. شکلها و جداول مربوط به ۵ موقعیت دیگر در ادامه آورده شده است.

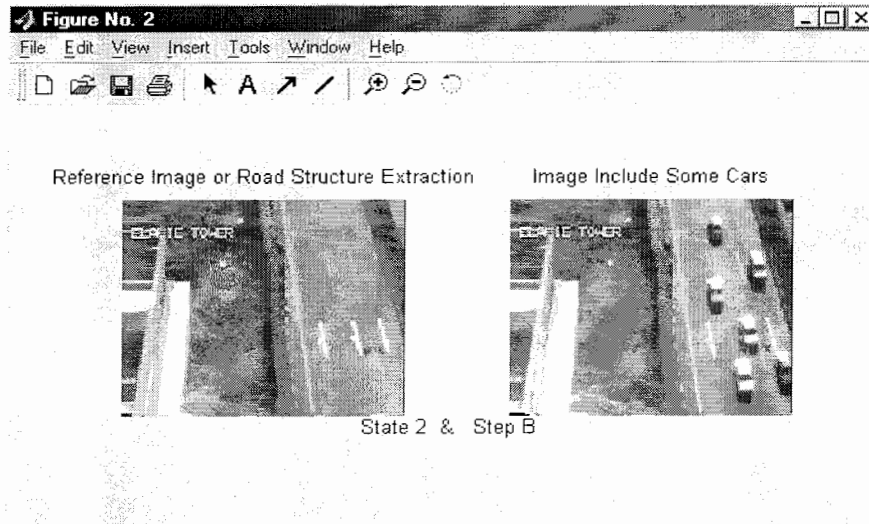
جدول ۵-۵ موقعیت دوم (B برج الهیه) و نتایج حاصل از تحلیل برنامه روی چهار عکس

نام فایل مرجع	نام فایل تصویر مورد تحلیل	تعداد واقعی	تعداد محاسبه شده	در صد خطای نسبی هر تصویر
emptyelahB.JPG	FullelahB1.JPG	6	7	14.2%
emptyelahB.JPG	FullelahB2.JPG	6.5	6.5	0%
emptyelahB.JPG	FullelahB3.JPG	5	4	20%
emptyelahB.JPG	FullelahB4.JPG	5	5	0%

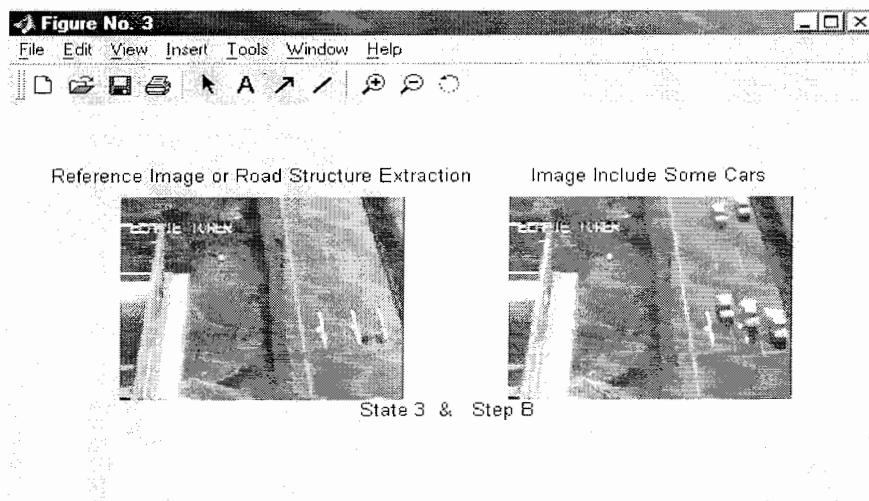
برای مشاهده تصاویر واقعی نیز در شکلهای بعدی به ترتیب ، تصاویر هر دو زوج فایل را برای هر بار تحلیل آورده ایم :



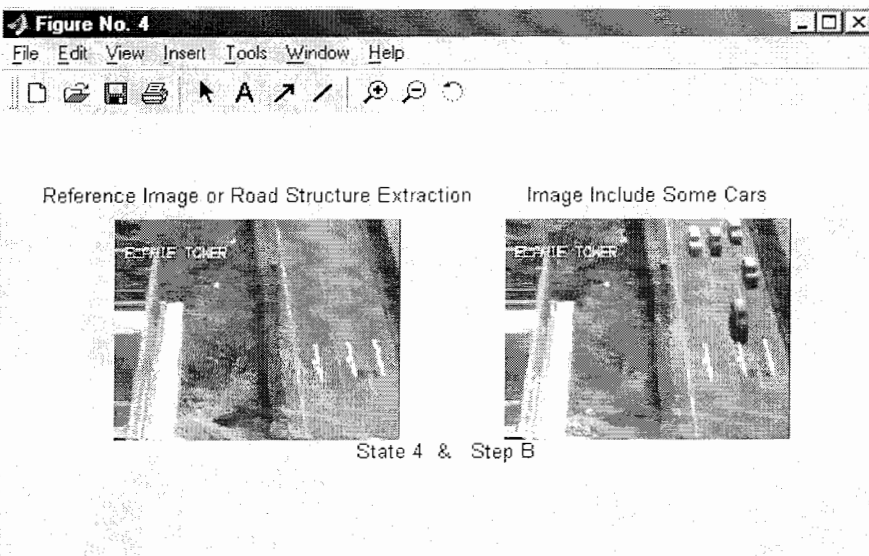
شکل ۵-۲۶ موقعیت دوم - تصویر (مرحله ۱)



شکل ۵-۲۷ موقعیت دوم - تصویر (مرحله ۲)



شکل ۵-۲۸ موقعیت دوم - تصویر (مرحله ۳)



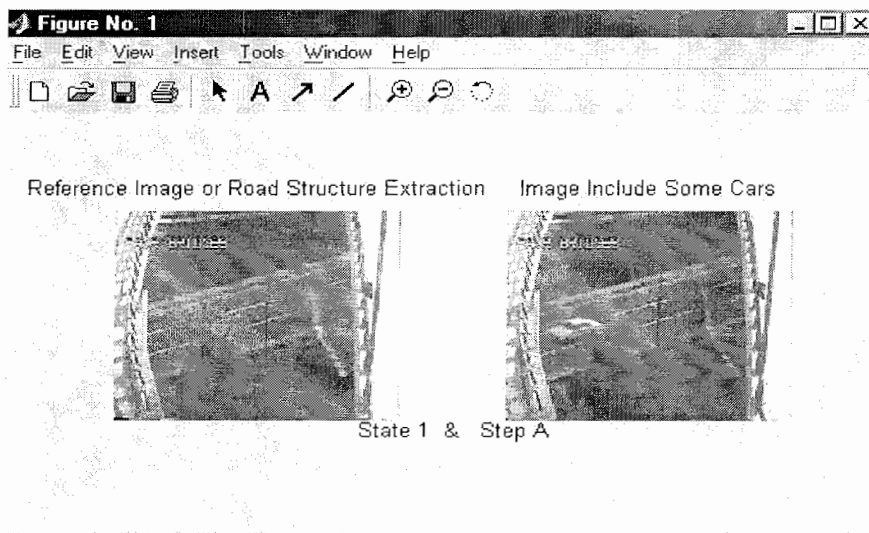
شکل ۵-۲۹ موقعیت دوم - تصویر (مرحله ۴)

باز هم همان طوری که مشهود است، در بین این چهار حالت در دو حالت هیچ خطایی نداریم، و در حالت دوم که مقدار خروجی ۶,۵ ذکر شده نیز جواب به علت این مقدار ۰,۵ دارد، که برنامه یک نیم خودرو را در تصویر شناسایی کرده است، که متعلق به باند مخالف است، و از آنجایی که دوربین مقداری از آن یکی باند را نیز پوشش داده چنین موردی پیش آمده است و در واقع برنامه در اینجایی خطا عمل کرده است. در دو مورد دیگر که برنامه خطا دارد، هر دو مربوط به همان عدم تناسب با سایر خودروهاست، یعنی در حالت اول وجود یک خودرو کشیده و در حالت سوم وجود یک خودروی کوچک در بالای تصویر باعث چنین خطاهایی شده است، که این خودروها بترتیب دوتا و هیچی شمرده شده اند، چون مورد اول کشیده و مورد دوم بیش از حد کوچک بوده است. احتمال شمرده نشدن همان طوری که در فصول قبل محاسبه شد، ۶,۲۵٪ می باشد، که در این مورد احتمال به وقوع پیوسته است.

بررسی موقعیت سوم و نتایجش در جدول ۵-۶ آمده است.

جدول ۵-۶ موقعیت سوم (پل فجر) و نتایج حاصل از تحلیل برنامه روی یک عکس

نام فایل مرجع	نام فایل تصویر مورد تحلیل	تعداد واقعی	تعداد محاسبه شده	در صد خطای نسبی هر تصویر
emptyfagrbrig.JPG	fagrbrig1.JPG	1	1	0%

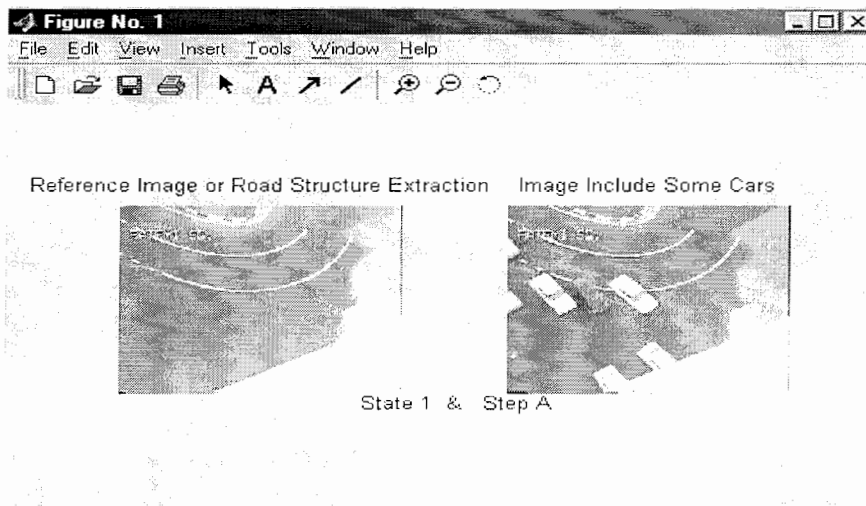


شکل ۵-۳۰ موقعیت سوم - تصویر (مرحله ۱)

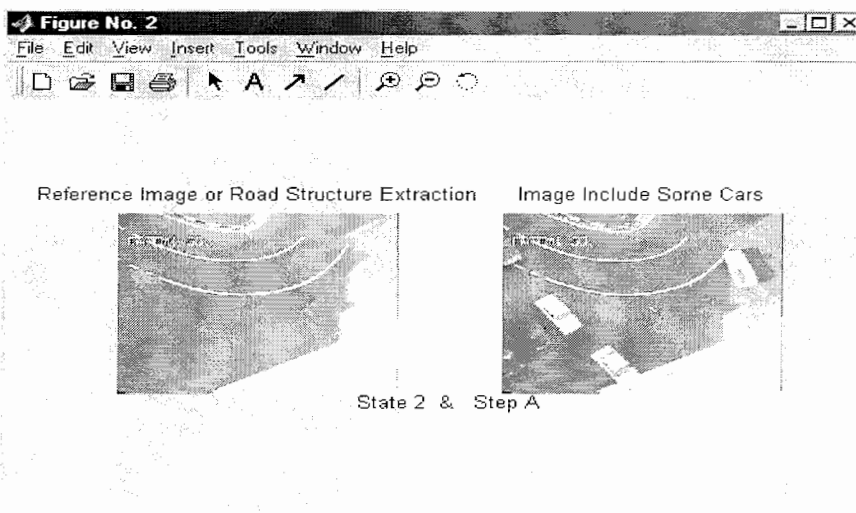
بررسی موقعیت چهارم و نتایجش در جدول ۵-۷ آمده است.

جدول ۵-۷ موقعیت چهارم (میدان فاطمی) و نتایج حاصل از تحلیل برنامه روی دو عکس

در صد خطای نسبی هر تصویر	تعداد محاسبه شده	تعداد واقعی	نام فایل تصویر مورد تحلیل	نام فایل مرجع
0%	6.5	6.5	Fatemisq1.JPG	emptyfatemisq.JPG
21%	9	7	Fatemisq2.JPG	emptyfatemisq.JPG



شکل ۵-۳۱ موقعیت چهارم - تصویر (مرحله ۱)



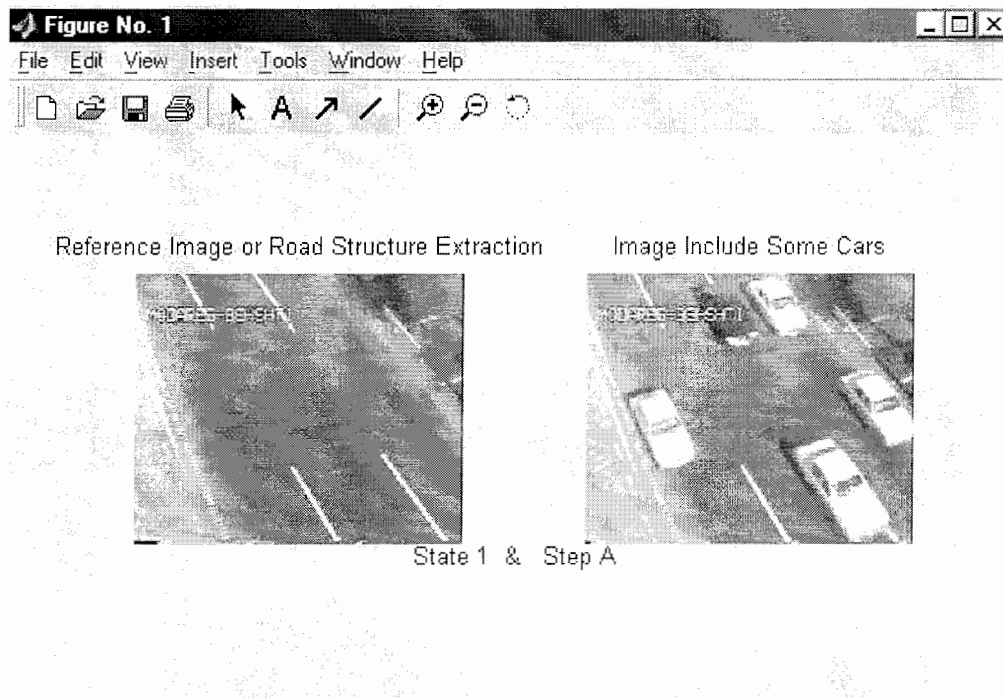
شکل ۵-۳۲ موقعیت چهارم - تصویر (مرحله ۲)

در موقعیت چهارم و حالت ۱ بازهم از آنجایی که ما کامل مسیر را پوشش نداده ایم، سه خودرو نصفه در تصویر مشاهده می شود. که سه شمارش ۰,۵ را در برنامه ایجاد می کنند و به همراه ۵ خودروی دیگر عدد ۶,۵ را از برنامه بدست می آوریم. پس برنامه در این حالت خطایی را ندارد. اما در حالت دوم از موقعیت چهارم به علت بی نظمی شدید ما مقداری خطا داریم، که چون بازهم هدف ما بزرگراهها با مسیرهای مستقیم است، اشکالی بر این روش وارد نیست.

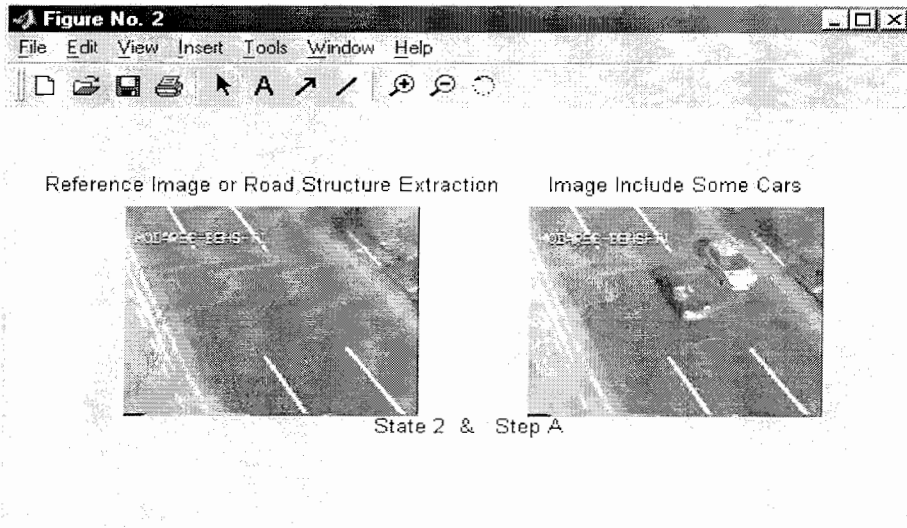
بررسی موقعیت پنجم و نتایجش در جدول ۵-۸ آمده است.

جدول ۵-۸ موقعیت پنجم (مدرس - بهشتی) و نتایج حاصل از تحلیل برنامه روی سه عکس

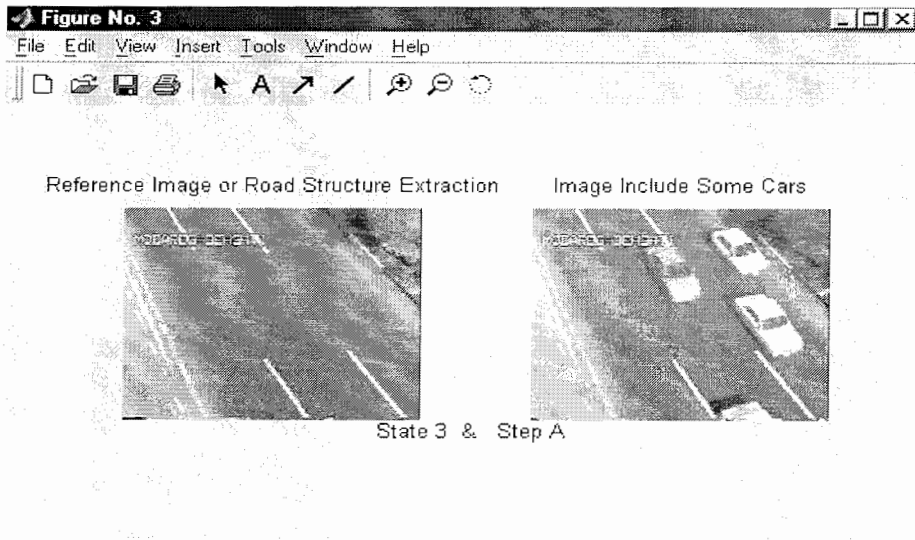
نام فایل مرجع	نام فایل تصویر مورد تحلیل	تعداد واقعی	تعداد محاسبه شده	در صد خطای نسبی هر تصویر
emptymodr-behesh.JPG	modr-behesh1.JPG	5	6	16.7%
emptymodr-behesh.JPG	modr-behesh2.JPG	2	2	0%
emptymodr-behesh.JPG	modr-behesh3.JPG	4	5	20%



شکل ۵-۳۳ موقعیت پنجم - تصویر (مرحله ۱)



شکل ۵-۳۴ موقعیت پنجم - تصویر (مرحله ۲)



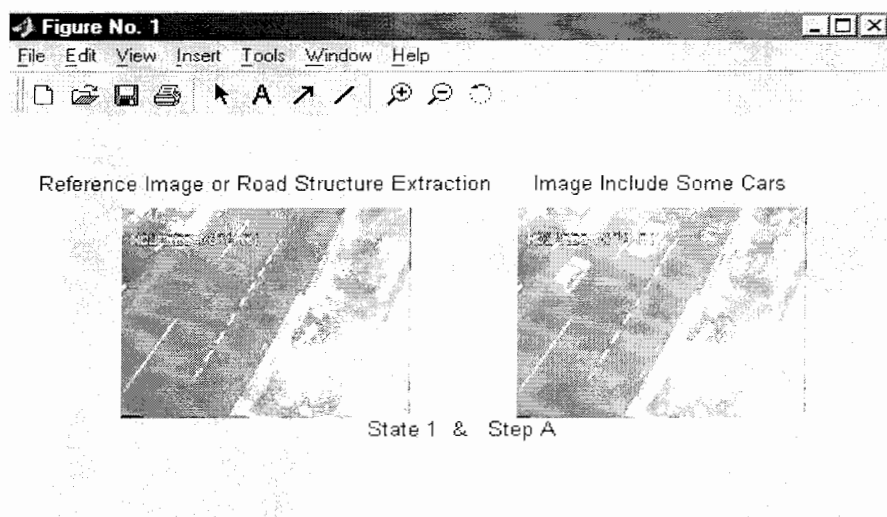
شکل ۵-۳۵ موقعیت پنجم - تصویر (مرحله ۳)

در موقعیت پنجم و حالت اول بازهم چون یک خودروی کشیده داریم، که آن دو بار شمرده شده و خطا ایجاد کرده است.

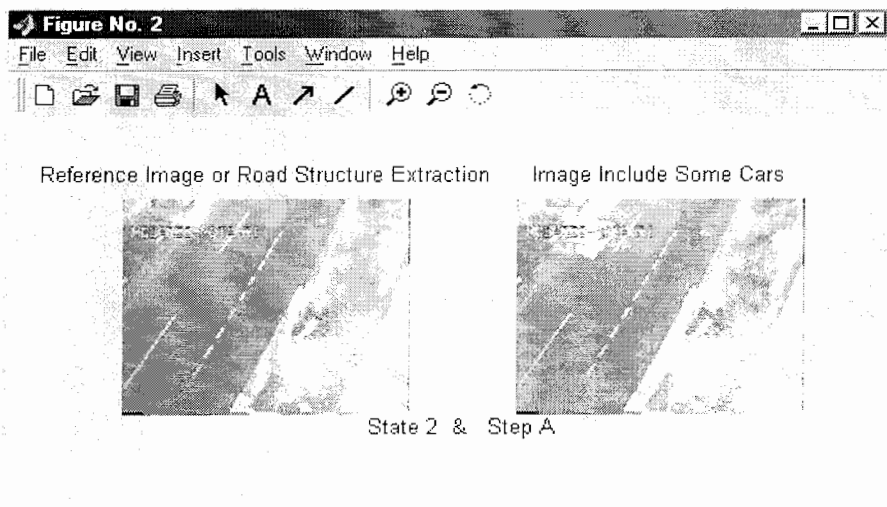
بررسی موقعیت ششم و نتایجش در جدول ۵-۹ آمده است.

جدول ۵-۹ موقعیت ششم (مدرس-مطهری) و نتایج حاصل از تحلیل برنامه روی دو عکس

در صد خطای نسبی هر تصویر	تعداد محاسبه شده	تعداد واقعی	نام فایل تصویر مورد تحلیل	نام فایل مرجع
0%	2	2	modr-motah1.JPG	emptymodr-motah.JPG
0%	1	1	modr-motah2.JPG	emptymodr-motah.JPG



شکل ۵-۳۶ موقعیت ششم - تصویر (مرحله ۱)



شکل ۵-۳۷ موقعیت ششم - تصویر (مرحله ۲)

پس از مشاهده این شش موقعیت و بررسی کامل آن در چند حالت ممکن برنامه ما از نظر درستی، در شمارش و کیفیت تحلیل در سطح مناسبی قرار گرفته است، زیرا میزان درستی نسبی حاصل از این ۲۲ مثال بطور متوسط ۹۱٫۹۳٪ می باشد، و درستی متوسط مربع برابر با ۹۳٫۰۸٪ می باشد که ما در کل خطای این روش را کمتر یا مساوی با ۱۰٪ بیان می کنیم.

توجه:

لازم به ذکر است که مقدار حد آستانه برای هر موقعیت پس از تنظیم ثابت شده است. حال برای اینکه ما در تمام طول روز بتوانیم این روش را پیاده سازی کنیم، می بایست حدود ۸ تصویر با میزان روشنایی های متفاوت را به عنوان مرجع محاسبات به برنامه اعمال کنیم. این کار با تاریک و روشن کردن تصویر مرجع توسط یکی از نرم افزارهای گرافیکی ممکن می شود. از جمله راهکارهای اعمال این ۸ تصویر مرجع به طور اتوماتیک اینست که ما با توجه به ساعت کامپیوتر مثلاً هر ۲ ساعت یک بار یکی از فایل های مرجع را فراخوانی کنیم، که این کار می تواند در ادامه این روش جدید انجام شود.

نتایج و پیشنهادات

نتیجه گیری:

افزایش حجم ترافیک وسایل نقلیه ، لزوم سیستم های پیشرفته تر کنترل ترافیک را ایجاب می- کند . اکثر کشورهای توسعه یافته دنیا از سالها پیش با احساس این نیاز ، بدنبال روش های بهینه تر ، دقیق تر و حتی الامکان خودکار برای کنترل ترافیک بوده اند . همراه با رایج شدن استفاده از سیستم های کنترل تلویزیونی برای نظارت و کنترل نقاط مهم وگره های ترافیکی ، خودکارسازی این امر بکمک پردازش تصویر مورد استفاده فراوانی پیدا کرده است .

بدین ترتیب لزوم وجود اپراتورهای انسانی برای نظارت روی تصاویر و نیز تشخیص و اعلام بموقع مواردی چون تراکم و تصادف از بین رفته و جای آن را سیستم های تشخیص وضعیت بکمک پردازش تصویر پر کرده است .

روش معرفی شده در این پایان نامه در همین راستا بوده و در واقع به نوعی یک شبیه سازی اپراتور ناظر وضعیت ترافیکی می باشد. کاربرد های مختلفی را می توان از این سیستم انتظار داشت ، برای مثال : پیاده سازی آن در خیابانها و چهارراهها ، بدین صورت که با نصب یک دوربین روی یک پل ، وضعیت ترافیک با دوربین فیلم برداری شده و پس از انجام عملیات پردازشی لازمه روی تصویر فوق ، علاوه بر کنترل زمان سبز و قرمز بودن چراغ های راهنمایی ، وضعیت ترافیکی موجود بررسی و استخراج شود و تحت عناوین ترافیک سبک ، متوسط(عادی) ، سنگین ، بسیار سنگین ، مسدود بر روی تابلوهای متغییر خبری ارسال شود تا رانندگان قبل از رسیدن به محل ترافیک بسیار سنگین مسیر خود را از راههای دیگر انحرافی انتخاب کنند ، تا به روند تعدیل ترافیک در خیابانها کمک گردد. از کاربردهای دیگر این سیستم ، نظارت بر بزرگراهها می باشد . بدین ترتیب که با نصب تعدادی دوربین ، در فواصل مناسب و در طول بزرگراهها ، می توان وضعیت ترافیکی را در هر لحظه در اختیار داشت و در صورت وجود ترافیک شدید یعنی تعداد بیش از حد ماشین ، یک هشدار(Alarm) ایجاد کرد .

همان طور که مشاهده شد ما این سیستم و روش جدید را روی چندین تصویر از ترافیک خودروها در حالات مختلف امتحان کردیم و حتی آن را بر روی دیگر اشیا امتحان نمودیم ، آنچه از کل این مطالب بر می آید ، آنست که ما با استفاده از این روش حتی در موارد بسیار نادر(از نظر بی نظمی) به درستی حدود ۹۰٪ می رسیم که با توجه به مزایا و کاربردهای متعدد این روش نتایج حاصله از آن بسیار مناسب و رضایت بخش است .

پیشنهادات:

۱- معرفی پروژه هایی که مطابق با آنچه در فصل دوم انجام شد، توابع بیشتری از پردازش تصویر را مورد تحلیل قرار دهد، تا مقایسه جامع تری مابین آنها انجام دهد.

۲- در فصل سوم ما تبدیلات زیادی را مطرح کردیم، ولی تمرکز اصلی کار را بر روی معرفی و نیز بیان طرز کار تبدیلاتی از جمله: فوریه و هارتلی و کسینوس گذاشتیم، که انجام تحقیق جامع - تر در مورد دیگر تبدیلات از جمله: Z و لاپلاس و..... و انجام کارهای نرم افزاری و عملی با این تبدیلات خالی از لطف نخواهد بود.

۳- در ادامه فصل سوم ملزومات سیستمهای تصویری را مطرح کردیم، یعنی با توجه به شرایطی که دستگاهها و سخت افزارهای موجود دارد، ما یک سری محدودیت جدی داریم که ما را به نوشتن و پیاده سازی برنامه ها و الگوریتمها و نرم افزارهای جدید که از هر نظر بهینه باشند، ترغیب می کند. اکنون که بازار تجارت الکترونیک و ارسال داده بسیار داغ می باشد، تولید هر گونه نرم افزاری حتی بسیار ساده و با کارایی جزئی میتواند هم از نظر اقتصادی مورد توجه قرار گیرد و هم گام مؤثری در جهت اعتلای قوای علمی بوده و هم سرافرازی و پیشرفت کشور عزیزمان ایران را بدنبال داشته باشد. تا ما هم سهم کوچکی، در قرار دادن کشورمان، در دستة پیشروها داشته باشیم.

از جمله پروژه های قابل تعریف در این فصل را میتوان:

۳-۱- معرفی روشی جدید و برتر نسبت به روشهای موجود تاکنون در مورد فشرده سازی اطلاعات، یعنی با توجه به شاخص های: حجم کم و کیفیت بالا، روش را به بهترین نحوی ارائه نماییم.

۳-۲- فشرده سازی تصاویر پزشکی با تبدیل هارتلی و بیان نتایج و نمایش کیفیت تصاویر حاصله.

۳-۳- معرفی روشهای بهتر و مستقل و یا معرفی روشهایی که از تلفیق روشهای دیگر کدینگ موجود بوده و عملیات کد کردن داده ها را انجام دهند.

البته با توجه به معیارهای ذکر شده در این فصل.

۴-۳- در انتهای فصل سوم ما کاربردهای دیگر تبدیل هارتلی از جمله:

۱- فشرده سازی صوت

۲- بازشناسی صوت و تصویر

۳- شناسایی سیستم و غیره.....

را نام بردیم، که میتوان هر یک را در اختیار یک گروه تحقیقاتی قرار داد. (چون پروژه های سنگینی است.) و یا به عنوان پروژه های چند نفره در دانشگاهها معرفی کرد، تا بر روی آنها کار شود. بدون شک، نتایج خوب و مناسبی خواهد داشت.

۴- در بخش دوم این پایان نامه، ما بحث کنترل ترافیک و شمارش خودروها را مطرح کردیم. با توجه به مشکلاتی که در مقدمه نیز آورده شده (گرانی دوربینها- نیازمند بودن به اپراتورهای زیاد و گرانی تراشه های DSP) میتوان روشهای دیگر و بهتری در مورد پردازش تصویر و کاربرد آن برای کنترل ترافیک ارائه داد.

۵- از جمله دیگر کاربردهای روش ارائه شده در شمارش تعداد هر شیء در تصویر است، که از آن میتوان برای دسته بندی و نیز شمارش قطعات (مخصوصاً قطعات و اشیاء یکسان) به طور اتوماتیک و شبانه روزی بهره برد و خطای معقولی هم داشت.

۶- همان طور که دیده شد، دید ما در مورد پردازش تصویر تقریباً با کلیه محققین و دانشجویانی که کارهایی در این زمینه ارائه کرده اند، متفاوت است. اکثر قریب به اتفاق آنها تاکید بر انجام چندین عملیات پردازشی در برنامه خود داشتند، که این امر چند مشکل ایجاد می کرد، از جمله:

۶-۱- چون برنامه ها پیچیده است و سرعتش پائین می باشد و نمی شود از آنها به طور بلادرنگ استفاده کرد، پس تنها به صورت یک کار تحقیقاتی صرف می ماند.

۶-۲- اگر هم آنها می خواستند از تراشه های DSP سرعت بالا استفاده کنند، یا آنقدر گران تمام می شد، که وجود حتی چند اپراتور اقتصادی تر بود و یا اینکه با پیشرفته ترین تراشه ها هم نمی

-توانستند بطور بلادرنگ کار را انجام دهند. در نتیجه کنترل ترافیک با این گونه روشها با کیفیت پایینی انجام می گرفت .

با توجه به این دو مشکل، ما تصمیم گرفتیم، دیدمان را تغییر داده و بجای استفاده از چندین عملیات پردازشی و پیچیده با ترکیب یکی، دو عملیات و استفاده از موقعیت، ارتفاع و زاویه دیدِ دوربین و نیز ابعاد متعارف خودروهای معمولی، این مهم را محقق کنیم.

۷- از همین دیدگاه که در قسمت ۶ پیشنهادها بیان شد، میتوان استفاده کرد و عملیاتهای مختلف پردازش تصویر را به آن تبدیل کرد. مثلاً: تشخیص نوع خودرو که تا کنون با برنامه های پیچیده صورت می گیرد، میتواند با شرایط معرفی شده در این پایان نامه انجام شود.

ضمیمه (۱)

برنامه جامع و کاربردی پردازش تصویر

```
%%% THIS PROGRAM IS WRITED FOR IMAGE PROCESSING(BRANCHS .....%%  
%%% ....BLUERRING & ADDING NOISE & CANCELING(NOISE&BLUER)WAYS&...%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% .....WORKING ON IMAGES.)) %%%%%%%%%  
I = imread('cameraman.tif');  
subplot(231);  
imshow(I);  
title('Original Image');  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
k=imresize(I,.7);  
L=imrotate(k,30);  
subplot(232);  
imshow(L);  
title('Resized&Rotated Image');  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
Rectcam=[93 13 81 69];  
Subcam=imcrop(I,Rectcam);  
subplot(233);  
imshow(Subcam);  
title('Cropped Image');  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
PSF = fspecial('gaussian',7,10);  
Blurred = imfilter(I,PSF,'symmetric','conv');  
subplot(234);  
imshow(Blurred);  
title('Blurred Image');  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
V = .001;
```

۱۰۰

```
        BlurredNoisy = imnoise(Blurred,'gaussian',0,V);

        subplot(235);

        imshow(BlurredNoisy);

        title('Blurred & Noisy');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% WAY1 (DECONVLUCY WITH PSF & ITERATE) %%%

        Shapmainway1= deconvlucy(Blurred,PSF);

        Shap1 = im2uint8(Shapmainway1);

        subplot(236);

        imshow(Shap1);

        title('Restored Image, NUMIT =15');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% WAY2 (DECONVLUCY & USEING Edgedtaper) %%%

        Edged=edgetaper(BlurredNoisy,PSF);

        Shap2=deconvlucy(Edged,PSF,5);

        figure;

        subplot(221);

        imshow(I);

        title('Original Image');

        subplot(222);

        imshow(Shap2);

        title('Edgedtaper effect');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% WAY3 (DECONVLUCY WITH Weight Array & To many ITERATE) %%%

        UNDERPSF = ones(size(PSF)-4);

        INITPSF = padarray(UNDERPSF,[2 2],'replicate','both');

        Weight = edge(I,'sobel',.3);

        se = strel('disk',2);

        Weight = 1-double(imdilate(Weight,se));

        Weight([1:3 end-[0:2]],:) = 0;

        Weight(:, [1:3 end-[0:2]]) = 0;

        subplot(223);
```

```
        imshow(Weight);

    title('Weight array');

        [J P] = deconvblind(Blurred,INITPSF,30,[],Weight);

    subplot(224);

    imshow(J);

    title('Restored Image WITH WEIGHT');

%%% WAY4(DECONVLUCY & USEING "NOISE POWER"+"Edgedtaper") %
    NP=V*prod(size(I));

        [reg1 LAGRA]=deconvreg(BlurredNoisy,PSF,NP*1.2);

    figure;

    subplot(221);

    imshow(I);

    title('Original Image');

        subplot(222);

    imshow(reg1);

    title('RESTORED WITH NP');

%%%%%%%%%% For obtaining better figure %%%%%%%%%%

        Edged1=edgetaper(reg1,PSF);

        Shap3=deconvlucy(Edged1,PSF,5);

    subplot(223)

    imshow(Shap3);

    title('RESTORED WITH NP+EDGE TAPER EFFECT');

%%% WAY5(DECONVLUCY & "USEING THE LAGRANGE MULTIPLIER") %%%

        Edged2=edgetaper(Blurred,PSF);

        reg2=deconvreg(Edged2,PSF,[],LAGRA/90);

    subplot(224)

    imshow(reg2);

    title('RESTORED WITH LAGRANGE MULTIPLIER');
```

```
%%%%%%%%%% WAY6(DECONVLUCY WITH "USEING INVERSE FILTER... %  
%%%%%%%%%% ...& CONTROLLING NOISE-SIGNAL-RATIO"(NSR) )%  
%%%%%%%%%%""""Inverse Filtering of Noisy Data"""":%%%%%%%%
```

```
noise = 0.15*randn(size(I));  
  
NSR = sum(noise(:).^2)/sum(im2double(I(:)).^2);  
  
Shap4= deconvwnr(BlurredNoisy,PSF,NSR);
```

```
figure;
```

```
subplot(221);
```

```
imshow(I);
```

```
title('Original Image');
```

```
subplot(222);
```

```
imshow(Shap4);
```

```
title('Restored with NSR');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Edged3=edgetaper(Shap4,PSF);
```

```
Shap5=deconvlucy(Edged3,PSF,5);
```

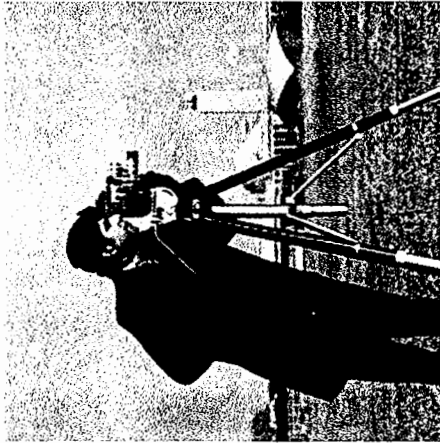
```
subplot(223)
```

```
imshow(Shap5);
```

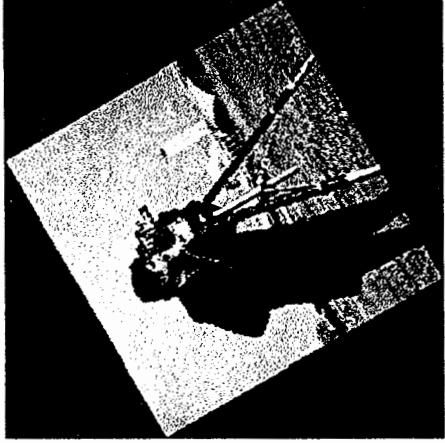
```
title('RESTORED WITH NSR+EDGE TAPER EFFECT');
```

```
%%%%%%%%%% FINISH %%%%%%%%%%
```

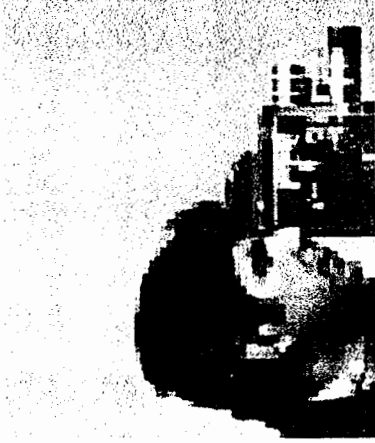
Original Image



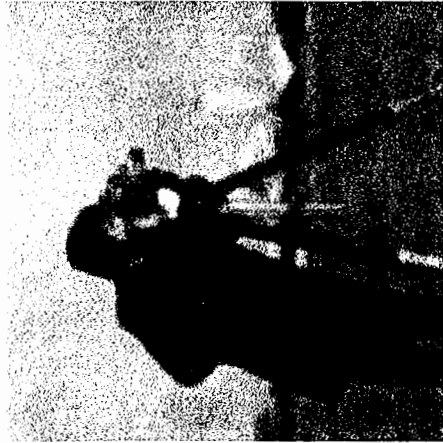
Resized&Rotated Image



Cropped Image



Blurred Image



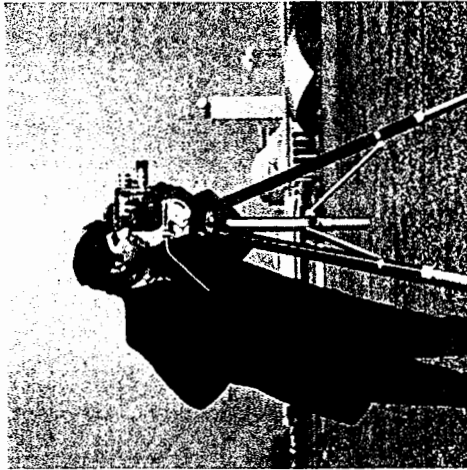
Blurred & Noisy



Restored Image, NUMIT =15



Original Image



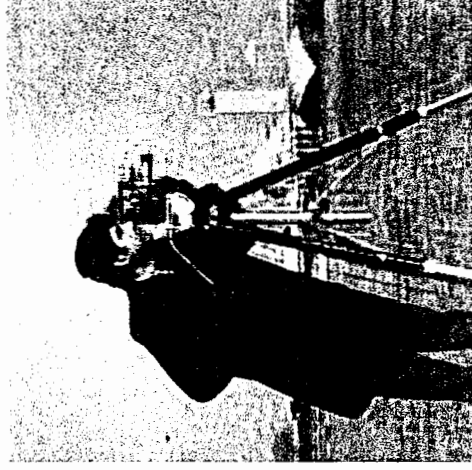
Edgedtaper effect



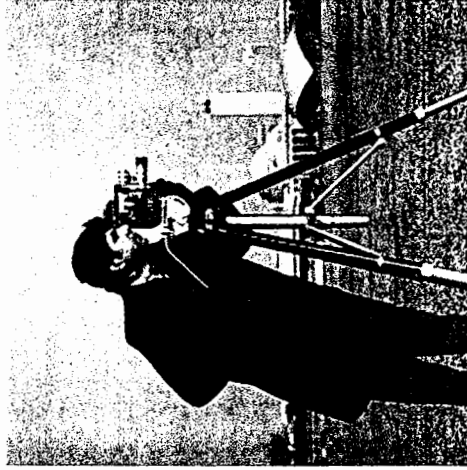
Weight array



Restored Image WITH WEIGHT



Original Image

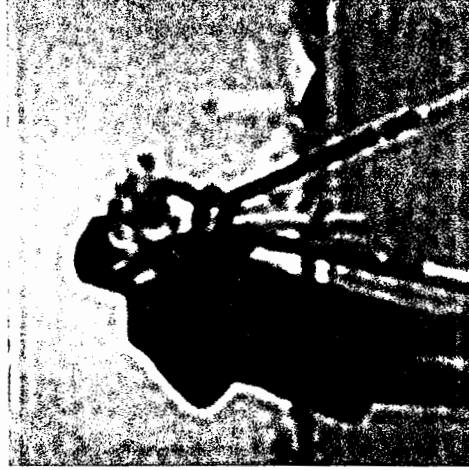


RESTORED WITH NP



1.1

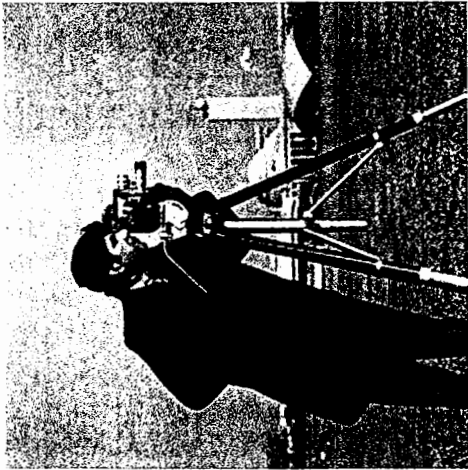
RESTORED WITH NP+EDGE TAPER EFFECT



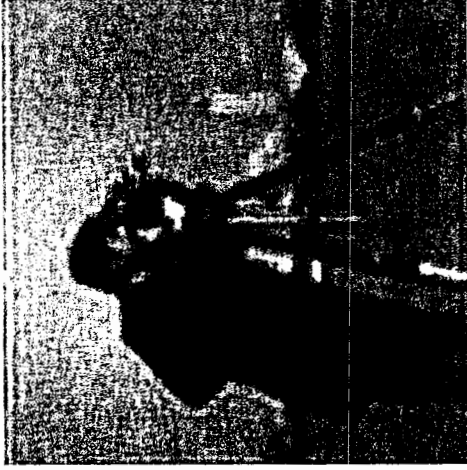
RESTORED WITH LAGRANGE MULTIPLIER



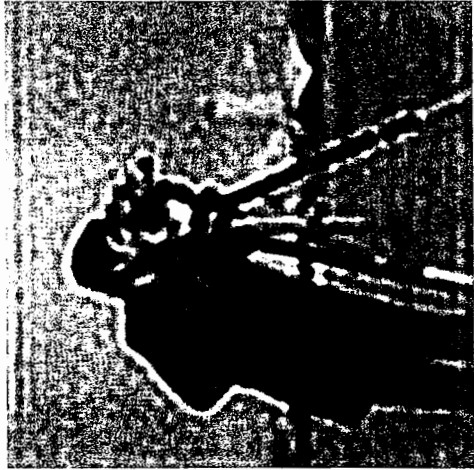
Original Image



Restored with NSR



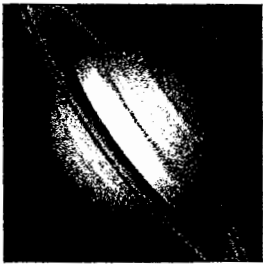
RESTORED WITH NSR+EDGE TAPER EFFECT



ضمیمه (۲)

فشرده سازی تصاویر به روش DCT با تعداد متغیر ضرایب تبدیل

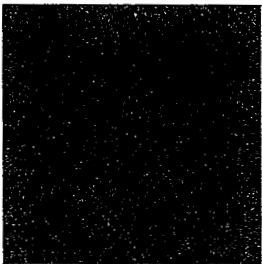
Original Saturn Image



Reconstructed Image

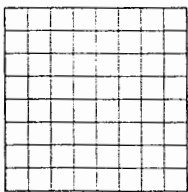


Error Image



The MSE (with images normalized) is 0.

DCT coefficients



64 Coefficients Selected 64

Apply

Select Image: Saturn

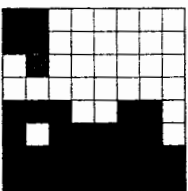
Info

Close

Original Saturn Image



DCT coefficients



32 Coefficients Selected

Reconstructed Image



Error Image



The MSE (with images normalized) is 0.000169.

Select Image:

Apply

Saturn

Info

Close

Original Saturn Image



Reconstructed Image

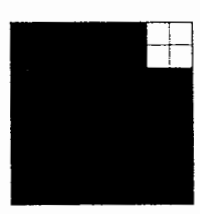


Error Image



The MSE (with images normalized) is 0.00203.

DCT coefficients



4 Coefficients Selected

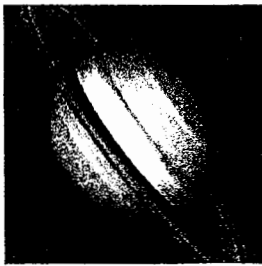
Select Image: Saturn

Apply

Info

Close

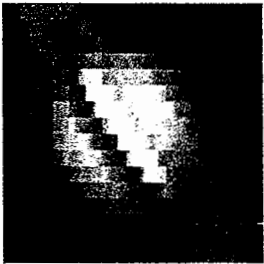
Original Saturn Image



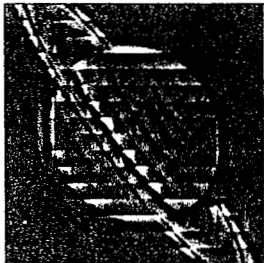
DCT coefficients



Reconstructed Image



Error Image



The MSE (with images normalized) is 0.00505.

2 Coefficients Selected 64

Apply

Select Image:

Saturn

Info

Close

ضمیمه (۳)

کدهای برنامه اثبات درستی منطق جدید در محیط *Visual Basic 6.0*

frmcarnumbers - 1

Option Explicit

'In example the refrence of the L1 is:
'10 10 250 10 10 250 10 10 250 10 10

Public Sub Cmdinitiallize_Click()

Dim mr(10) As Integer
Dim mx(10) As Integer
Dim ma(10) As Integer
Dim x() As String
Dim v() As String

Dim z As String
Dim y As String
Dim c As Integer
Dim b As Integer
Dim i As Integer
Dim u As Integer

y = txtin1.Text

x() = Split(y)

For i = LBound(mr) To UBound(mr)

mr(i) = CInt(x(i))

Next i

z = txtin2.Text

v() = Split(z)

For i = LBound(ma) To UBound(ma)

ma(i) = CInt(v(i))

Next i

'mr(0) = 10

'mr(1) = 10

'mr(2) = 250

'mr(3) = 10

'mr(4) = 10

'mr(5) = 250

'mr(6) = 10

'mr(7) = 10

'mr(8) = 250

'mr(9) = 10

'mr(10) = 10

For i = LBound(ma) To UBound(ma)

u = ma(i) - mr(i)

If Abs(u) > 20 Then

mx(i) = 1

Else

mx(i) = 0

End If

Next i

frmcarnumbers - 2

```
        c = 0
For i = LBound(mx) To UBound(mx) - 1
    If mx(i) <> mx(i + 1) Then
        c = c + 1
    End If
Next i
        b = c / 2
txtout1.Text = "The Number Of Car(s) In The One Line is: " & " " & b
End Sub

Public Sub Cmdexit_Click()
    End
End Sub
```

ضمیمه (۴)

کدهای برنامه شمارش ماشین ها در یک خط به همراه خروجی

%%%%%%%%%% Calculation for one line of picture %%%%%%%%%%%

%%%%%%%%%%load picture%%%%%%%%%%
clc

clear

Imageinput1 = imread('cameraman.tif');

Imageinput2 = imread('rice.tif');

imshow(Imageinput2)

%%%%%%%%%Difine variables%%%%%%%%%

numcarsline1=0;edge=0;

xref1=zeros(1,255);

xline1=zeros(1,255) ;

xres1=zeros(1,255) ;

diffpixels=zeros(1,255);

Rectcam1=[0 30 255 0];

Rectcam2=[0 40 255 0];

xref=imcrop(Imageinput1,Rectcam1);

xline=imcrop(Imageinput2,Rectcam2);

for a=1:255

xref1(1,a)=xref(a);

end

for a=1:255

xline1(1,a)=xline(a);

end

for a=1:255

diffpixels(a) = xref1(1,a)-xline1(1,a);

if abs(diffpixels(a))>90

```
        xres1(1,a)=1;

    end

end

for a=1:1:254

    if xres1(1,a)~= xres1(1,a+1)

        edge=edge+1;
    end

end

xref

. xline

    diffpixels

        xres1

            numcarsline1=edge/2
```

```
>>
xref =

Columns 1 through 19
 158  159  163  157  158  155  163  159  158  158  162  162  161  164
163  166  162  166  158

Columns 20 through 38
 165  165  168  164  162  164  165  164  162  165  161  162  164  162
167  166  166  169  169

Columns 39 through 57
 167  168  168  170  173  168  169  171  170  168  176  169  166  173
169  173  176  174  170

Columns 58 through 76
 171  170  176  172  173  171  175  170  169  173  172  174  173  171
169  172  175  173  176

Columns 77 through 95
 172  171  174  179  174  175  172  171  174  176  181  179  179  175
178  176  176  177  178

Columns 96 through 114
 178  179  178  178  178  178  181  181  181  181  180  178  179  179
182  183  183  183  186

Columns 115 through 133
 188  181  184  182  186  184  184  185  185  190  184  185  185  182
180  181  182  182  183

Columns 134 through 152
 182  185  186  185  184  184  184  184  186  186  185  185  186  185
183  185  184  188  180

Columns 153 through 171
 183  184  179  187  183  179  181  178  179  176  177  177  177  179
180  179  178  179  177

Columns 172 through 190
 177  177  177  175  178  179  179  175  174  171  175  171  171  172
171  174  173  172  171

Columns 191 through 209
 168  168  170  171  170  170  169  169  169  168  170  168  168  162
170  170  165  167  164

Columns 210 through 228
 164  166  166  165  162  166  162  161  162  165  159  160  161  158
160  162  161  161  159

Columns 229 through 247
 162  163  161  161  160  154  159  152  156  158  158  155  155  155
157  154  150  157  155
```

Columns 248 through 255

156 155 152 152 155 154 148 155

xline =

Columns 1 through 19

166 171 175 180 180 185 185 183 183 180 173 162 146 128
110 101 96 93 93

Columns 20 through 38

92 93 92 92 92 93 93 92 92 93 92 93 92 92
92 93 93 101 118

Columns 39 through 57

141 161 173 173 169 168 162 166 165 162 161 163 159 163
158 145 131 114 96

Columns 58 through 76

94 93 93 93 93 93 93 93 93 93 93 93 93 93
93 93 93 93 93

Columns 77 through 95

93 93 93 93 93 93 93 93 96 110 125 141 144 141 129
110 99 93 93 93

Columns 96 through 114

93 93 93 93 93 105 124 146 161 161 158 160 163 162
158 155 133 118 103

Columns 115 through 133

93 93 93 93 93 93 93 101 123 143 163 168 173 179
177 177 183 184 186

Columns 134 through 152

180 171 155 131 107 93 92 93 92 94 101 115 138 157
170 177 179 181 185

Columns 153 through 171

189 195 195 194 194 190 188 188 188 186 180 157 136 115
98 94 93 93 99

Columns 172 through 190

110 117 128 128 121 112 103 96 93 91 91 91 91 90
91 90 90 90 90

Columns 191 through 209

91 90 90 90 90 90 94 110 128 151 162 165 166 168
170 174 175 180 181

Columns 210 through 228

183 185 188 190 192 193 190 185 185 183 182 180 179 180
183 182 183 183 180

Columns 191 through 209

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

Columns 210 through 228

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

Columns 229 through 247

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

Columns 248 through 255

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

numcarsline1 =

4

>>

ضمیمه (۵)

کدهای برنامه شمارش کل ماشین ها در یک تصویر

به همراه یک مثال ، از دیگر کاربردهای این روش

```
%%%%%%%%%%The main of program as function %%%%%%%%%%%  
  
function y= tez3(k)  
  
%%%%%%%%%%load picture%%%%%%%%%%  
  
Imageinput = imread('rice.tif');  
  
%imshow(Imageinput)  
  
%%%%%%%%%%Difine variables%%%%%%%%%%  
  
numcarsline1=0;edge=0;  
  
xref1=zeros(1,255);  
  
    xline1=zeros(1,255) ;  
  
        xres1=zeros(1,255) ;  
  
            diffpixels=zeros(1,255);  
  
Rectcam1=[0 30 255 0];  
  
    Rectcam2=[0 k 255 0];  
  
xref=imcrop(Imageinput,Rectcam1);  
  
    xline=imcrop(Imageinput,Rectcam2);  
  
for a=1:255  
    xref1(1,a)=xref(a);  
  
end  
  
for a=1:255  
    xline1(1,a)=xline(a);  
  
end  
  
for a=1:255  
    diffpixels(a) = xref1(1,a)-xline1(1,a);  
  
    if abs(diffpixels(a))>20  
        xres1(1,a)=1;  
  
end
```

```
end

for a=1:1:254

if xres1(1,a)~= xres1(1,a+1)

    edge=edge+1;
end

end

%numcarsline1

y=edge/2;

%-----
```

%%%%%%%%%% Calculation All Cars %%%%%%%%%%%

clc

clear

Totalnumcar=0;

for k=2:36:254

 numcarsline1= tez3(k);

 Totalnumcar=Totalnumcar+ numcarsline1;

end

disp('The Number Of Car(s) In The picture is:')

Totalnumcar

Im= imread('rice.tif');

imshow(Im)

%%%%%%%%%The main of program as function %%%%%%%%%%

function y= tez3rice(k)

%%%%%%%%%load picture%%%%%%%%%

Imageinput1 = imread('cameraman.tif');

Imageinput = imread('blood1.tif');

Rectcam=[0 0 255 255];

Imageinput2=imcrop(Imageinput,Rectcam);

%imshow(Imageinput2)

%%%%%%%%%Difine variables%%%%%%%%%

numcarsline1=0;edge=0;

xref1=zeros(1,255);

xline1=zeros(1,255) ;

xres1=zeros(1,255) ;

diffpixels=zeros(1,255);

Rectcam1=[0 30 255 0];

Rectcam2=[0 k 255 0];

xref=imcrop(Imageinput1,Rectcam1);

xline=imcrop(Imageinput2,Rectcam2);

for a=1:255

xref1(1,a)=xref(a);

end

for a=1:255

xline1(1,a)=xline(a);

end

for a=1:255

diffpixels(a) = xref1(1,a)-xline1(1,a);

```
        if abs(diffpixels(a))>39
            xresl(1,a)=1;
        end
    end

    for a=1:1:254
        if xresl(1,a)~= xresl(1,a+1)
            edge=edge+1;
        end
    end

    %numcarsline1
    y=edge/2;

    %-----
```



```
%%%%%%%%%% Calculation All Cars %%%%%%%%%%%  
  
clc  
  
clear  
  
Totalnumcar=0;  
  
for k=2:39:254  
  
    numcarsline1= tez3rice(k);  
  
    Totalnumcar=Totalnumcar+ numcarsline1;  
  
end  
  
disp('The Number Of Car(s) In The picture is:')  
  
    Totalnumcar  
  
Im= imread('blood1.tif');  
  
imshow(Im)
```

>> The Number Of Car(s) In The picture is:

Totalnumcar =

37.5000

>>

ضمیمه (٦)

برنامه حل مثالهای اولیه با تصاویر واقعی از بزرگراه

%%%%%%%%%The main of program as function %%%%%%%%%

```
function y= tez3realst(k)

%%%%%%%%%load picture%%%%%%%%%

path(path, 'C:\WINDOWS\Desktop\pol havaeil')

Imageinput1 = imread('emptyA','jpg');
Imageinput2 = imread('fullA','jpg');

%%%%%%%%%Difine variables%%%%%%%%%
numcarsline1=0;edge=0;

xref1=zeros(1,255);

xline1=zeros(1,255) ;
xres1=zeros(1,255) ;

diffpixels=zeros(1,255);
Rectcam1=[0 k 255 0];
Rectcam2=[0 k 255 0];

xref=imcrop(Imageinput1,Rectcam1);
xline=imcrop(Imageinput2,Rectcam2);

for a=1:255
    xref1(1,a)=xref(a);
end

for a=1:255
    xline1(1,a)=xline(a);
end

for a=1:255
    diffpixels(a) = xref1(1,a)-xline1(1,a);
    if abs(diffpixels(a))>100
        xres1(1,a)=1;
    end
end

for a=1:1:254
    if xres1(1,a)~= xres1(1,a+1)
        edge=edge+1;
    end
end
```

end

end

y=edge/2;

%-----finish-----

%%%%%%%%%% Calculation All Cars %%%%%%%%%%%

clc

clear

Totalnumcar=0;

for k=1:100:192

numcarsline1= tez3realst(k);

Totalnumcar=Totalnumcar+ numcarsline1;

end

disp('The Number Of Car(s) In The picture is:')

Totalnumcar

path(path, 'C:\WINDOWS\Desktop\pol havaei1')

Imageinput1 = imread('emptyA','jpg');

Imageinput2 = imread('fullA','jpg');

figure;

subplot(1,2,1);

imshow(Imageinput1);

title('Reference Image or Road Structure Extracti

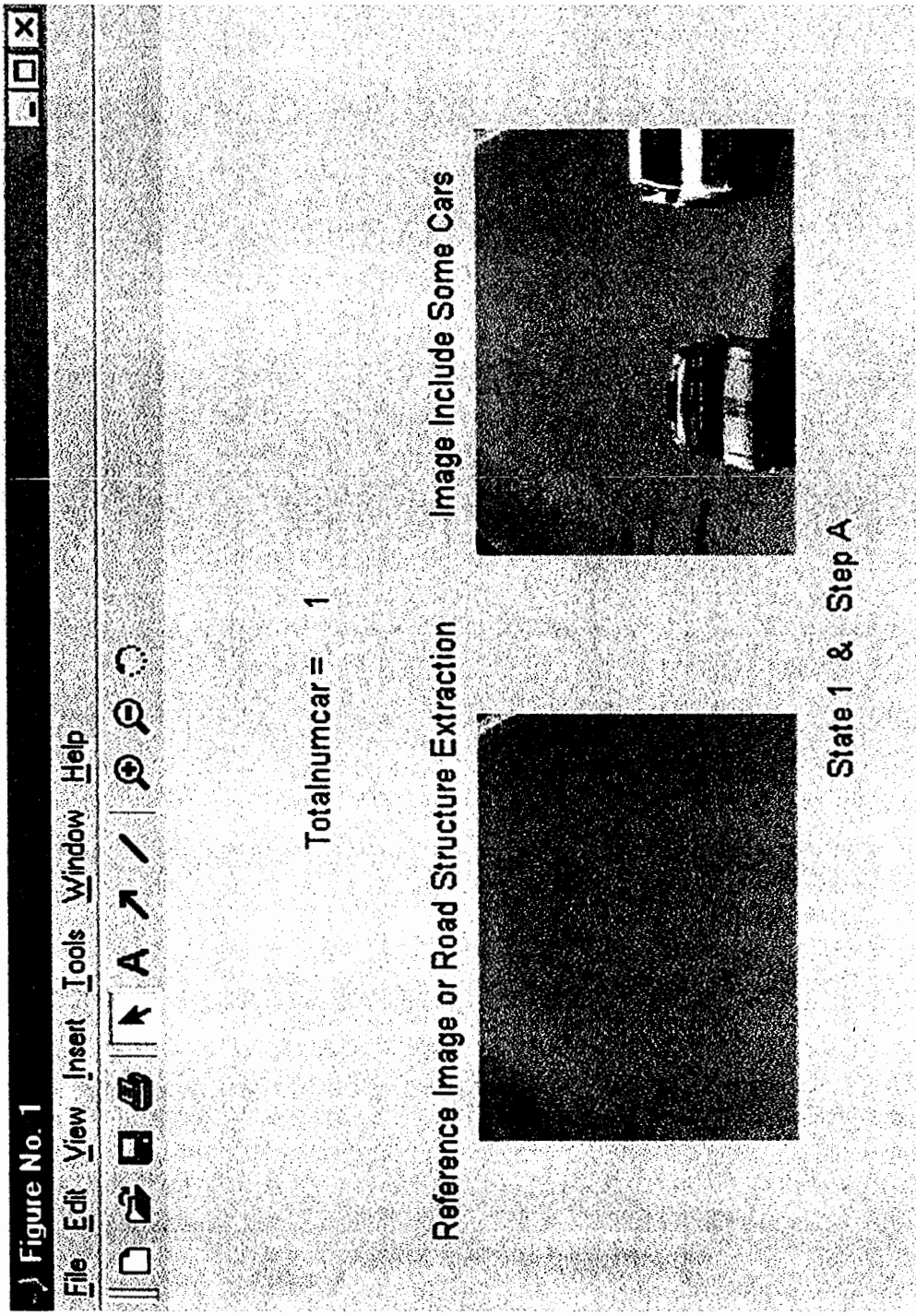
on')

subplot(1,2,2);

imshow(Imageinput2);

title('Image Include Some Cars')

text(-144 ,220,' State 1 & Step A ')



فهرست منابع و مآخذ

مراجع بخش اول - پردازش تصویر

مراجع فصل اول :

- 1- Van trees , Harry L. Detection , estimation and modulation theory .Massachusetts Institute of Technology.Wiley & Sons,Inc ,1971.
- 2- Gonzalez , R.C. & Wintz ,P. Digital image Processing . Addison-Wesley Publishing ,1987.
- 3-Papoulis , A. Probability & random variables & stochastic-processes , Mc Graw-Hill,1965 .

مراجع فصل سوم :

- 4- Gonzalez , R.C. & Wintz , P. loc.Cit .
- 5- Oppenheim , Alan & Willsky , S. & Young , Ian T. Signals & systems . Prentice Hall ,1983 .
- 6- Pitas , I . Digital image processing algorithms and applications . Wiley & Sons,Inc , 2000 .
- 7- Prabhu , k.M.M. An efficient radix-3 FHT algorithm , *IEEE* 13th conference on digital signal processing , 1997.
- 8- Yeh , Panshu . 1989. Data compression properties of the Hartley Transform . *IEEE* transaction on acoustics,Speech and signal processing ,vol.37, No.3, pp .110-132 .
- 9- Castelman , K. Digital image processing . Wiley & Sons,Inc.1994.
- 10- Cappellini ,V. Time-varying image processing & moving object . Mc Graw-Hill . 1991.

- 11- Inigo , R .1989.Application of machine vision to traffic monitoring & control .
IEEE transactions on vehicular technology ,vol.38 , pp.112-122 .
- 12- Hoose , N . Computer image processing in traffic engineering . Research
Studies Press , Taunton . 1991.
- 13- Koto, K. & Nishiyama , K. & Deguchi , M. & Ito,Y. & Sukai , K .Image-processing
traffic flow measuring system of the Hokuriku Express way . Tokyo , 1998 .
- 14- Mitsui , A. & Aoki ,M .Preceding vehicle detection using horizontal edge and
uniform region .Proceedings of the Asian conference on computer vision , Osaka ,
Japon , 1993 , 31-34 .
- 15- Sobottka ,K. & Wetzel, D. 1997. Attention control integrated in a system to
Autonomous driving & collision avoidance under egomotion . Bavarian Research
Center . *IEEE*,pp. 796-800 .
- 16- Abut , H. & Bilgin , A. & Bernardi, R.M . 1995. A case study in IVHS
implementation an image processing application for 1-15 hov lanes . E.C.E
department , San Diego State University , San Diego ,CA92182, *IEEE*,pp . 2979-
2982 .
- 17- Tan ,T.N. & Sullivan , G.D. & Baker, K.D. 1996. Efficient image gradient-based
object localization & recognition .Computer Science , University Of Reading
England , *IEEE* , pp . 397-402 .
- 18- Soh , J. & Chun,B.T. & Wang,M.1995. Analysis of road image sequences of vehicile
counting .Systems Engineering Research Institute . Yoosung ,Korea . *IEEE*, pp .
679-683 .
- 19- Noll ,D. & Werner, M.1996. Real-Time vehicile tracking & classification . Bochum ,
Germany . *IEEE*, pp . 101-106 .

مراجع فصل پنجم :

۲۰- بلاتر، دیوید و فلاشیمن، گلن و راث، استیون اصول و مبانی استفاده از اسکنر و پردازش

- تصویر، ترجمه خوش کیش، رضا، چاپ اول، تهران، کانون نشر علوم، ۱۳۷۸.

۲۱- فکور، علی آشنایی با زبان برنامه نویسی Matlab، چاپ اول، مشهد، جهاد دانشگاهی

مشهد، ۱۳۷۱.

۲۲- کوثری، علیرضا آموزش زبان برنامه نویسی ویژوال بیسیک ۶، چاپ دوم، تهران، موسسه

فرهنگی هنری دیباگران تهران، ۱۳۸۱.

۲۳- مختاری، موهند و ماری، میشل و صمدی بخارایی، وحید کاربرد Matlab & simulink در

مهندسی، چاپ اول، انتشارات خراسان، ۱۳۸۰.

24- Awcock, G.W. & Thomas, R. Applied image processing. Mc Graw-Hill. 1995.

واژه نامه فارسی به انگلیسی

Detector	آشکار ساز
Increase	افزایش
Noise	اعوجاج
Transfer	انتقال
Reflection	انعکاس
Real time	بلادرنگ
Image processing	پردازش تصویر
Edge detection	تشخیص لبه
Description	توصیف
Threshold	حد آستانه
Inductive loop	حلقه مغناطیسی
Frequency domain	حوزه فرکانس
Time domain	حوزه زمان
Vehicle	خودرو
Binary	دودویی
Digital Camera	دوربین دیجیتال
Gray level	سطح خاکستری
Simulation	شبیه سازی
Off-line	غیر لحظه ای (نا همزمان)

On-line	لحظه ای (همزمان)
Averaging	متوسط گیری
Comparator	مقایسه کننده
Frame rate	نرخ تصویربرداری
Pixel	نقطه (پیکسل یا جز تصویر)
Block diagram	نمودار بلوکی
Up dating	نوسازی
Resolution	وضوح



Shahrood University of Technology

College of Electrical Engineering

Topic:

***Counting Vehicles In High Way With
Using Image Processing Methods***

By:

Seyed Saeed Mirsharifi

A Thesis is submitted in partial fulfillment of the
requirement for the degree of Master Of Science in
Electrical Engineering .

Adviser:

Dr.H.Toosian Shandiz

Nov 2003

Abstract

This thesis includes two sections. The first section, discusses about fundamentals & advantages & applications & operations of operators of **Digital Image Processing**. This section is completed with representation two programs about these topics (Appendix (1,2)). This section includes three chapters:

- 1- Reconstructing noisome images
- 2- Description of image processing software
- 3- Transforms algorithms & images compressing.

In the second section, we describe a computer vision system, according to digital image processing methods to count vehicles on roads & streets. The traffic images are captured by a video camera and digitized in to a computer. The measuring algorithm is based on comparison between a calculated reference without vehicles and the current image of traffic lanes. Test under real traffic conditions were satisfactory, with over **90%** of accuracy.

Key words: Digital image processing, traffic control, instrumentation, counting vehicles.