

بهبود هم‌زمان دقت و سرعت شناسایی مانع در بینایی استریوی قطع شده با داده‌های التراسونیک در خودروهای هوشمند

فرشاد غلامی^۱، اسماعیل خان‌میرزا^۲، محمد ریاحی^۳

چکیده

امروزه ترافیک به چالشی برای همه تبدیل شده است. یکی از راه‌های عبور از این مسئله، هوشمندسازی خودروهاست. لذا در این پژوهش به ارتقای ادراک محیط با استفاده از ترکیب داده‌های التراسونیک و بینایی استریو پرداخته شده است. در این پژوهش محققان پیشنهاد داده‌اند که روش بینایی استریوی قطع شده با استفاده از داده‌های التراسونیک به نحوی به‌روزرسانی شود که دقت و سرعت شناسایی مانع در خودروهای هوشمند به‌صورت هم‌زمان افزایش پیدا کند. بنابراین علاوه بر شباهت نور در پنجره‌ی انطباق، عمق پیکسل‌های همسایه به‌گونه‌ای مورد استفاده قرار گرفته است که بدون افزایش نمایی بار محاسباتی، اهداف مذکور محقق گردد. در همین راستا دو نوع پنجره‌ی انطباق تعریف گردیده است. یکی از این دو نمونه پنجره انطباق، مشابه با روش بینایی استریو قطع شده و دیگری به‌عنوان پنجره استنتاج است. با استفاده از پنجره استنتاج پیشنهادی بر پایه‌ی بینایی استریوی قطع شده، امکان اعمال تأثیر عمق پیکسل‌های همسایه به میزان کافی و تأثیرگذار فراهم شده و سبب کاهش ۶۱/۵۷ درصدی خطا گردیده است. در ادامه با توجه به نحوه‌ی پیاده‌سازی بر روی کارت گرافیک، علاوه بر کاهش خطا، سرعت شناسایی مانع ۴۳/۹۳ درصد بهبود پیدا کرده است. بهبودهای مذکور سبب می‌شود که چنانچه خودرویی نیاز به شناسایی محیط در هر یک متر از حرکت خود داشته باشد، بتواند به سرعت ۱۷۸٫۱ km/h دست پیدا کند و سبب افزایش قابلیت اطمینان در خودروهای هوشمند گردد.

کلید واژه‌ها

خودروی هوشمند، شناسایی مانع، بینایی استریو، التراسونیک

۱- مقدمه

به مسئله‌ی ترافیک می‌توان به‌صورت یک مسئله جمعی نگاه می‌شود لذا راه‌حلی‌هایی که در این دیدگاه مطرح می‌شود، به این صورت است که با توزیع بهینه بار ترافیکی بر روی جاده‌های

مختلف، مدت زمان سفر مجموعه‌ی خودروها را کاهش داده و ترافیک روان‌تری ایجاد شود. در همین راستا می‌توان به پژوهش‌های ونگ و همکاران اشاره کرد. ایشان با استفاده از الگوریتم ژنتیک موفق به توزیع بار ترافیک شده‌اند [۱].

در نگاه دیگری که به مسئله‌ی ترافیک می‌شود، سعی در کنترل و تصمیم‌گیری هر خودرو به‌صورت فردی دارد. در واقع محققین با تلاش برای حذف عوامل انسانی که سبب کاهش بهینگی حرکت خودروهاست، سعی در بهبود ترافیک و امنیت مسافران دارند. در همین راستا با توجه به دسته‌بندی که توسط کندول و همکاران ارائه گردیده است، پژوهش‌های این دیدگاه در دسته‌بندی زیر دنبال شده است [۲]:

- لایه‌ی ادراک
- لایه‌ی تصمیم‌گیری

این مقاله در دی‌ماه ۱۴۰۱ دریافت، در اسفندماه بازنگری و سپس پذیرفته شد.

^۱ دانشجوی دکتری مهندسی مکانیک، دانشگاه علم و صنعت ایران، تهران

رایانامه: f_gholami@mecheng.iust.ac.ir

^۲ دانشیار، مهندسی مکانیک، دانشگاه علم و صنعت ایران، تهران

رایانامه: khanmirza@iust.ac.ir

^۳ استاد، مهندسی مکانیک، دانشگاه علم و صنعت ایران، تهران

رایانامه: riahi@iust.ac.ir

نویسنده مسئول: اسماعیل خان‌میرزا

dorl.net/dor/20.1001.1.23831197.1402.10.3.4.6

● لایه‌ی کنترل

باتوجه به دسته‌بندی مذکور یکی از مهم‌ترین لایه‌هایی که در ایجاد یک خودروی هوشمند مورد توجه است، لایه‌ی ادراک است. چراکه ادراک اشتباه سبب تصمیم‌گیری اشتباه و در نهایت کنترل خودرو برای دستیابی به هدفی اشتباه خواهد شد. برای ایجاد ادراک در خودروهای هوشمند دو دسته حسگر مورد استفاده قرار می‌گیرد:

- حسگرهای فعال
- حسگرهای غیرفعال
- حسگرهای ترکیبی

۱-۱- حسگرهای فعال

از این نوع حسگرها می‌توان به حسگر تعیین مسافت با استفاده از نور^۱ و حسگرهای زمان پرواز^۲ اشاره کرد. با استفاده از حسگرهای فعال می‌توان با دقت و صحت بالا و در بازه‌ی زمانی بسیار کوتاهی به ادراک از محیط دست پیدا کرد. در همین راستا می‌توان به تحقیقات ژی و همکاران اشاره کرد که سعی در ایجاد ادراکی کامل از خروجی حسگر لیزر-راداری داشته‌اند [۳]. با وجود تمام مزایایی که استفاده از این حسگرها در خودروهای هوشمند همواره دارد، معایبی نیز با خود بر مجموعه تحمیل می‌کنند، که از آن‌ها می‌توان به قیمت بالا و تأثیرپذیری آن‌ها از نمونه‌های مشابه در محیط عملکردی یکسان نام برد. باتوجه به توضیحات داده‌شده محققان به دنبال استفاده از حسگرهای دیگری رفته‌اند که از نمونه‌های آن می‌توان به دوربین‌ها تصویربرداری اشاره کرد. روش‌های مختلف استفاده از این نوع حسگرها در ادامه توضیح داده شده است.

۱-۲- حسگرهای غیرفعال

یکی از مهم‌ترین حسگرهای غیرفعال که در خودروهای هوشمند مورد استفاده قرار می‌گیرد، دوربین‌های تصویربرداری نوری است. از این نوع حسگر به شکل‌های مختلفی در خودروهای هوشمند بهره برده شده است. شکل‌های مختلف استفاده از دوربین تصویربرداری در دو حوزه‌ی چیدمان و الگوریتم‌های شناسایی توسعه پیدا کرده‌اند.

چیدمان تک‌چشمی: در این نوع چیدمان تنها یک دوربین تصویربرداری استفاده می‌شود. نمونه‌هایی از استفاده از این آرایش در ایجاد ادراک محیط، از حرکت دوربین به صورت عمود بر راستای تصویربرداری حاصل می‌شود که به آن روش ساختار بر پایه‌ی حرکت^۳ گفته می‌شود. از این دسته از پژوهش‌ها می‌توان به پژوهش کائو و همکاران اشاره کرد. ایشان با استفاده از ساختار بر پایه‌ی حرکت به ایجاد تجسم سه‌بعدی از محیط پرداخته‌اند [۴]. باتوجه به این‌که در این نوع از آرایش، راستای تصویربرداری عمود

بر راستای حرکت دوربین است، امکان استفاده از آن در خودروهای هوشمند برای شناسایی موانع روبروی خودرو نیست. چیدمان چند چشمی: این نوع چیدمان در واقع الهام گرفته شده از بینایی موجودات زنده است که به آن بینایی استریو گفته می‌شود. در این چیدمان باتوجه به میزان اختلاف بین دو تصویر، موقعیت نقاط متناظر با پیکسل‌های تصاویر در محیط سه‌بعدی محاسبه می‌گردد. برای نمونه می‌توان به کیم و همکاران اشاره کرد که از بینایی استریو برای ایجاد ادراک در ربات کشاورزی استفاده کرده‌اند [۵] و یا وو و همکاران از بینایی استریو برای کنترل ربات جراح بهره برده‌اند [۶]. اگرچه استفاده از بینایی استریو ایرادات مطرح شده در روش‌های پیشین را ندارد، اما از دو منظر دارای چالش است:

● کیفیت پاسخ

● حجم بالای محاسبات

در همین راستا محققین برای افزایش دقت و سرعت ایجاد ادراک از محیط با استفاده از این حسگرها روش‌های مختلفی را ارائه کرده‌اند. این روش‌ها به چهار دسته‌ی کلی تقسیم می‌شوند:

● روش‌های محلی

● روش‌های جهانی

● روش‌های ویژگی پایه

● روش‌های مبتنی بر یادگیری

در روش‌های محلی به منظور پیدا کردن عمق هر پیکسل در محیط تنها به شدت نور پیکسل‌هایی که در همسایگی وجود دارند توجه می‌شود. در این دسته از پژوهش‌ها محققین به ارائه‌ی پنجره‌های انطباق مختلف و یا تابع هزینه‌های مختلف پرداخته‌اند [۷-۹]. برای مثال لیو و همکاران به منظور افزایش کیفیت پاسخ‌ها به ارائه پنجره‌هایی با تطبیق‌پذیر پرداخته‌اند [۱۰]. محققینی نیز به استفاده از چند کاندید پنجره انطباق روی آورده‌اند [۱۱-۱۳]. ایشان با ایجاد چند فرم پنجره‌ی انطباق سبب بهبود انطباق‌ها شده‌اند. اگرچه موارد مذکور بهبودهایی بر روی خروجی بینایی استریو گذاشته اما محققینی نیز به دنبال بهبود هرچه بیشتر پاسخ‌ها بودند. لذا علاوه بر روش‌های پیشین که به روش‌های محلی انطباق گفته می‌شد، روش‌های انطباق جهانی^۴ نیز به وجود آمده است [۱۴-۱۶].

در روش‌های انطباق جهانی علاوه بر شباهت شدت نور در پیکسل‌های همسایگی، فرض نرمی نیز لحاظ گردیده است. در همین راستا محققینی به استفاده از این روش برای بهبود هرچه بیشتر تجسم سه‌بعدی محیط بهره برده‌اند [۱۶، ۱۷]. اگرچه این روش سبب بهبود نتایج حاصله از بینایی استریو گردیده است، اما بار محاسباتی که آن را نسبت به روش‌های محلی بیش از پیش دوچندان کرده است. به نحوی که مرتبه‌ی محاسبات از $O(n)$ به O^3 افزایش پیدا می‌کند. لذا محققینی به ارائه روش‌هایی برای حل

¹ LIDAR

² Time of flight

³ Structure from motion

⁴ Global

بودن داده‌هایی که به واسطه‌ی آن‌ها آموزش انجام می‌شود، وابسته بوده است. این وابستگی سبب کاهش قابلیت اعتماد این روش‌ها می‌گردد، بنابراین در این پژوهش از این روش استفاده نگردیده است.

۱-۳- حسگرهای ترکیبی

استفاده از حسگرهای فعال و غیرفعال به صورت تکی سبب می‌شود که ویژگی‌های مثبتی در مجموعه‌ی شناسایی موانع تضمین گردد، اما با توجه به توضیحات داده‌شده، استفاده تکی آن‌ها عیوبی را نیز به مجموعه تحمیل می‌کند که سبب کاهش سرعت و یا دقت نتایج می‌گردد. در همین راستا برخی از محققین، به استفاده‌ی هم‌زمان چند نمونه از حسگرهای فعال و غیرفعال به صورت هم‌زمان روی آورده‌اند. آن‌ها با استفاده از داده‌های این حسگرها به صورت هم‌زمان سبب افزایش دقت و یا سرعت مجموعه‌ی ترکیبی گردیده‌اند. برای مثال ژو و همکاران به ترکیب داده‌های حسگر اینرسی، دوربین تصویر برداری و حسگر تعیین مسافت با استفاده از نور پرداخته‌اند [۳۰]. همچنین ژانگ و همکاران با استفاده از سه داده‌ی برداشت شده از حسگر لیزری، موقعیت نقاط کلیدی در تصاویر را تخمین زده‌اند [۳۱]. در ادامه گریتر و همکاران پژوهش ژانگ و همکاران را با ترکیب تصویر و حسگر تعیین مسافت با استفاده از نور انجام داده‌اند [۳۲] و یا به پژوهش غلامی و همکاران اشاره کرد که با استفاده از داده‌های حسگر التراسونیک توانسته‌اند محدوده‌ی جستجو در بینایی استریوی محلی را کاهش داده و سبب افزایش سرعت شناسایی موانع در مجموعه‌ی نهایی گردند [۳۳]. با توجه به توضیحات داده‌شده، پژوهشگران حاضر به ارتقای روش پیشنهادی در مطالعه‌ی پیشین غلامی و همکاران از دو جنبه‌ی سرعت و دقت پرداخته‌اند که در ادامه روش پیشنهادی تشریح و نتایج آن گزارش گردیده است.

۲- الگوریتم

به منظور تشریح روش پیشنهادی در ابتدا ساختار مورد استفاده برای ایجاد داده‌ها توضیح داده شده است و پس از آن نوآوری‌های پیشنهادی در دو حوزه‌ی افزایش دقت و افزایش سرعت تشریح گردیده است.

۲-۱- تولید داده‌ها

با توجه به توضیحات مقدمه، با استفاده از ترکیب داده‌های حسگرهای مختلف می‌توان معایب استفاده‌ی هر یک از حسگرها به صورت تکی را پوشش داد. برخلاف حسگرهای بینایی استریوی، استفاده از حسگر التراسونیک به صورت تکی موقعیت موانع در عرض محیط را فراهم نمی‌کنند. از طرفی بینایی استریوی با وجود قابلیت‌هایی که دارد، بار محاسباتی بالایی داشته و استفاده از آن به صورت زمان واقعی را دچار چالش می‌کند. با استفاده از هم‌زمان این دو نوع حسگر می‌توان پس از شناسایی عمق موانع در محیط

بینایی استریوی به فرم جهانی پرداخته‌اند. ایشان به ارائه تغییراتی در روش قطع گراف پرداخته‌اند [۱۸، ۱۹]. با توجه به بار محاسباتی بالایی که این دسته از روش‌ها دارند، در این پژوهش از این نوع روش‌ها استفاده نگردیده است بلکه ایده‌ی آن به صورت کاهش یافته و متناسب با بینایی استریوی قطع شده با داده‌های التراسونیک پیشنهاد و مورد استفاده قرار گرفته است.

روش‌های دیگری که از آن‌ها برای انجام فرایند انطباق استفاده می‌شود، روش‌های ویژگی پایه است. در این دسته از روش‌ها با استفاده از ویژگی خاصی که در محیط وجود دارد به انجام فرایند انطباق پرداخته می‌شود. نمونه‌هایی از این ویژگی‌ها که در پژوهش‌ها مورد استفاده قرار گرفته است، شامل، رنگ قرمز موجود در چراغ عقب خودروها، تقارن و یا رنگ سفید خطوط جاده‌ها و یا لبه‌ها است [۲۰-۲۲]. اگرچه این نوع روش‌ها سبب افزایش سرعت انطباق گردیده است، اما با توجه به احتمال وجود نقض کننده‌ی ویژگی‌های مذکور در محیط، استفاده از این ویژگی‌ها سبب کاهش قابلیت اعتماد مجموعه‌ی حاصله می‌گردد. لذا در این پژوهش از این دسته از روش‌ها استفاده نگردیده است.

دسته‌ی دیگری از روش‌هایی که برای فرایند انطباق استفاده می‌شود، روش‌های مبتنی بر یادگیری است. در این روش‌ها با استفاده از یک فرایند یادگیری، مجموعه‌ای را به وجود می‌آورند که با انطباق ورودی‌های جدید با آنچه با آن آموزش داده شده است، سعی در پیدا کردن نقاط کلیدی در دو تصویر و سپس انطباق آن‌ها دارد [۲۳، ۲۴].

روش‌هایی که بر پایه‌ی شناسایی موانع با استفاده از یادگیری ایجاد شده است، در دو دسته‌ی یادگیری سطحی و یادگیری عمیق تقسیم می‌شود. محققان نمونه‌های مختلفی از یادگیری سطحی برای دسته‌بندی محیط نظیر ماشین بردار پشتیبان^۱، تقویت کننده‌ی فوقی^۲ و یا شبکه عصبی با یک یا دو لایه مخفی با یادگیری دارای نظارت را ایجاد کرده‌اند [۲۵]. همچنین محققان ترکیبی از روش‌های مذکور با مواردی نظیر هیستوگرام شیب‌های جهت‌دار^۳ پرداخته‌اند [۲۶]. با این وجود روش‌های ایجاد شده بر مبنای یادگیری سطحی برای ایجاد ارتباط بین متغیرهای چندگانه در مسائل دارای ابعاد بالا مناسب نیست [۲۷]. با توجه به موارد مذکور روش‌های یادگیری عمیق مورد توجه قرار گرفته است که از نمونه پژوهش‌ها می‌توان به روش نگوین و همکاران اشاره کرد که با استفاده از یادگیری عمیق به شناسایی موانع پرداخته‌اند [۲۸]. لازم به ذکر است که نتایج آن‌ها محدود به محیط دوبعدی بوده است. همچنین رومس و همکاران نیز با این روش به شناسایی موانع پیش‌بینی نشده پرداخته‌اند [۲۹]. بر خلاف نتایج قابل امیدوارکننده، استفاده از این روش‌ها همواره در محیط‌های ناشناخته دارای چالش بوده است و نتایج حاصله به شدت به غنی

¹ Support Vector Machine

² Adaptive Boost(AdaBoost)

³ Histogram of Oriented Gradient(HOG)

$$dis = \frac{f \times B}{\Delta} \quad (1)$$

که در رابطه‌ی (۱) عبارت Δ بیان‌گر فاصله‌ی مانع تا دوربین‌ها (فاصله‌ی شناسایی شده توسط حسگر التراسونیک) و f بیان‌گر فاصله‌ی کانونی دوربین‌ها و B بیان‌گر فاصله دو دوربین از یکدیگر و dis لایه‌ی شناسایی مانع هستند. لازم به ذکر است باتوجه‌به استفاده از دوربین ایده‌آل در محیط شبیه‌سازی، فرایند کالیبراسیون برای حذف اعوجاج از تصاویر حذف گردیده است.

۲-۲- افزایش کیفیت پاسخ

فرایند انطباق در بینایی استریوی محلی شامل پیدا کردن میزان انحراف در میان تمام لایه‌های انطباق باتوجه‌به کمینه تابع هزینه در بین تمام انحراف‌های ممکن است. با این حال در فرایند انطباق در روش بینایی استریوی محلی قطع‌شده به‌ازای هر پیکسل تنها به پیدا کردن کمینه از بین سه مقدار خلاصه می‌گردد. این سه مقدار متناظر با تابع هزینه به‌ازای سه لایه‌ی زیر است:

۱. لایه‌ی شناسایی شده توسط حسگر التراسونیک
۲. لایه‌ی پیشین لایه‌ی شناسایی شده
۳. لایه‌ی پسین لایه‌ی شناسایی شده

لذا تعداد محاسباتی که به‌ازای هر پیکسل انجام می‌شود برابر با رابطه‌ی (۲) است.

$$\begin{aligned} \text{NumberOfCalculation} \\ = (\text{numOfRowsOfKernel}) \\ \times (\text{numOfColumnOfKernels}) \times 3 \end{aligned} \quad (2)$$

که در رابطه‌ی فوق منظور از عبارت Kernel، پنجره انطباق است.

کاهش تعداد لایه‌های شناسایی مانع، سبب به‌وجود آمدن پاسخ‌های اشتباه در میان پاسخ‌های صحیح می‌گردد که متناظر با کمینه‌های محلی است که حالا به‌عنوان کمینه‌های جهانی شناخته شده‌اند. به‌منظور افزایش کیفیت پاسخ می‌توان از روش انطباق متقاطع بهره برد. اگر فرایند انطباق متقاطع نیز انجام شود تعداد محاسبات ارائه‌شده در رابطه‌ی (۲) برابر خواهد شد با رابطه‌ی (۳):

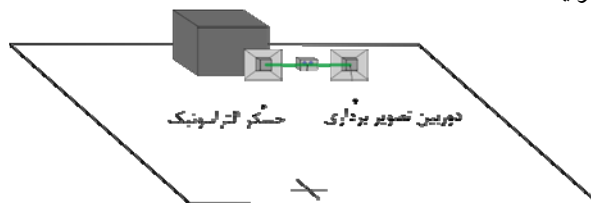
$$\begin{aligned} \text{NumberOfCalculation} = \\ = (\text{numOfRowsOfKernel}) \\ \times (\text{numOfColumnOfKernels}) \times 3 \times 2 \end{aligned} \quad (3)$$

حال اگر اندیس پیکسل لنگری در تصویر مرجع (تصویر راست) در فرایند انطباق برابر با (i, j) باشد، آنگاه اندیس‌های پیکسل لنگری که در فرایند انطباق استفاده می‌شود شامل موارد ذکر شده در جدول (۱) خواهد بود (در صورت استفاده از انطباق قطع‌شده و دارای انطباق متقاطع).

لازم به ذکر است که در جدول ۱ عبارت i ، j و dis به ترتیب بیانگر شماره سطر، ستون و لایه‌ی شناسایی شده توسط حسگر التراسونیک است. لازم به ذکر است که تعداد سطرهای یکتا در جدول ۱ برابر ۵ سطر است.

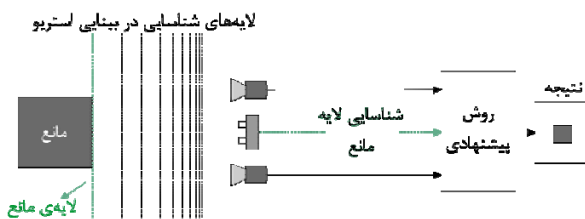
توسط حسگر التراسونیک، فرایند انطباق در بینایی استریو را محدود کرده و با کاهش چشمگیری که در بار محاسباتی به وجود می‌آید، سبب نزدیک شدن سرعت پردازش به یک فرایند زمان واقعی شد، بنابراین داده‌ها محصول مجموعه‌ای مطابق با شکل ۱ است.

به‌منظور کاهش عوامل جانبی در داده‌های معرفی شده به روش پیشنهادی و همچنین امکان مقایسه‌ی نتایج آن با پژوهش‌های پیشین از داده‌های تولیدشده در پژوهش [۳۳] استفاده شده است. داده‌های اشاره‌شده با استفاده از مجموعه‌ای شامل یک حسگر التراسونیک و دو دوربین تصویربرداری در نرم‌افزار پلندر تولید شده است. دوربین‌های تصویربرداری به فاصله‌ی ۲۰ سانتی‌متر و در ارتفاع ۲ متری از سطح زمین و حسگر التراسونیک در میان آن‌ها قرار گرفته است. دلیل انتخاب این ارتفاع از سطح زمین استفاده این نوع سیستم‌ها در خودروهای هوشمند سنگین و همچنین مجموعه‌های نصب شده بر روی خودروهای هوشمند سواری برای بهبود دید از جاده بوده است. فاصله‌ی کانونی دوربین‌ها نیز ۵۰ میلی‌متر انتخاب شده است. دلیل انتخاب این فاصله‌ی کانونی نزدیک بودن خروجی دوربین‌ها به آنچه انسان می‌بیند بوده است. حسگر التراسونیک نیز به صورت مجازی در سیستم ایجاد گردیده است به نحوی که فاصله‌ی موانع تا ۵۰ متر را با دقت ۱ سانتی‌متر تولید کند.



شکل (۱): نمای از محیط مجازی شبیه‌سازی در نرم‌افزار پلندر

همانند روش بینایی استریوی قطع‌شده با داده‌های التراسونیک [۳۳] پس از شناسایی مانع توسط حسگر التراسونیک، لایه‌ای که مانع در آن وجود دارد به الگوریتم تحویل داده می‌شود و الگوریتم پیشنهادی باتوجه‌به داده‌ی دریافتی و تصاویر استریو به شناسایی مانع می‌پردازد. این فرایند در شکل ۲ نمایش داده شده است.



شکل (۲): نمای از فرایند شناسایی مانع در روش پیشنهادی در فرایند شناسایی لایه‌ی مانع با در اختیار داشتن فاصله‌ی مانع و فاصله‌ی کانونی دوربین‌ها و با استفاده از رابطه‌ی (۱) لایه‌ی شناسایی مانع و یا انحراف^۱ در فرایند بینایی دوچشمی تعیین می‌گردد.

¹ Disparity

هزینه‌ی آن محاسبه گردیده است که باتوجه به ابعاد تصاویر، قابل چشم پوشی است.

باتوجه به توضیحات داده شده بدون افزایش بار محاسباتی کیفیت پاسخ بهبود پیدا خواهد کرد و عملاً تعدادی از پاسخ ها که در کمینه‌ی محلی بودند حذف گردند. در پژوهش پیشین پس از انجام بینایی استریو قطع شده با انطباق متقابل، با استفاده از فیلتر میان گیر در فرایند پس پردازش استفاده می گردیده است که بار محاسباتی آن برابر با رابطه‌ی (۴) خواهد بود که محققان در این پژوهش، این مقدار از محاسبات را حذف کرده و راه حلی با کارایی بیشتر در تکمیل روش قبلی ارائه کرده اند.

$$\text{FilterCalculation} = (\text{NumOfAllPixels}) \times \text{numOfRowOfKernel} \times \text{numOfColumnOfKernel} \quad (4)$$

در روش های بینایی استریوی محلی فرایند انطباق تنها با استفاده از شباهت شدت نور در پیکسل های موجود در همسایگی یک پیکسل در تصویر مرجع با تعدادی پیکسل در تصویر دوم انجام می شود. اما در نمونه‌ی دیگری از بینایی استریو که به بینایی استریو جهانی مشهور است، علاوه بر شباهت شدت نور در پیکسل های همسایگی به عمق متناظر به پیکسل های همسایگی در محیط نیز تکیه می شود که به آن عبارت نرمی^۱ گفته می شود. باتوجه به توضیحات داده شده، تابع هزینه مورد استفاده در این حالت به صورت رابطه‌ی (۵) خواهد بود.

$$\text{CostFunction} = \text{LocalCostFunction} + \text{SmothnessCost} \quad (5)$$

در روش های جهانی تعداد تابع هزینه هایی که محاسبه می شود به صورت زیر خواهد بود.

$$\text{numOfCostCalculation} = (\text{NumOfAllPixels}) \times \text{numOfDisparity} \times \text{KernelSize} \quad (6)$$

که در عبارت فوق KernelSize برابر است با:

$$\text{KernelSize} = \text{numOfRowsOfKernel} \times \text{numOfColumnOfKernel} \quad (7)$$

دلیل این حجم از محاسبه، این است که در این روش پوش تولیدی به تعداد $(\text{NumOfAllPixels}) \times \text{numOfDisparity}$ حالت امکان ایجاد دارد. در این روش پوشی که به صورت ۳ بعدی و به یک باره تشکیل می شود. در واقع پاسخ یکی از حالت های موجود در بین موارد امکان پذیر است که باتوجه به تابع هزینه های محاسبه شده، به یک باره حاصل خواهد شد. در شکل (۳) نمایی از حالت های موجود به نمایش گذاشته شده است.

حال اگر روش بینایی استریو قطع شده را محدود به ۳ لایه پیشین کنیم تعداد محاسبات برابر خواهد شد با رابطه‌ی (۸):

$$\text{numOfCostCalculation} = (\text{NumOfAllPixels})^3 \times \text{KernelSize} \quad (8)$$

جدول (۱): اندیس پیکسل های لنگری در تصاویر چپ و راست در فرایند انطباق قطع شده و دارای انطباق متقاطع

اندیس تصویر چپ	اندیس تصویر راست
$i, j + \text{dis} - 1$	i, j
$i, j + \text{dis}$	i, j
$i, j + \text{dis} + 1$	i, j
$i, j + \text{dis}$	$i, j - 1$
$i, j + \text{dis}$	i, j
$i, j + \text{dis}$	$i, j + 1$

همچنین اگر موقعیت پیکسل لنگری در توابع هزینه برای فرایند انطباق قطع شده و بدون انطباق متقاطع به ازای پیکسل های (i, j) و $(i, j + 1)$ و $(i, j + \text{dis})$ در تصویر مرجع راست استخراج گردد، متناظر با اندیس های ارائه شده در جدول های (۲) الی (۴) خواهد بود. با مقایسه جدول های (۱) تا (۴) مشابهت هایی بین المان های جدول ها (۲) الی (۴) با جدول (۱) دیده خواهد شد.

جدول (۲): اندیس های پیکسل لنگری مورد استفاده برای محاسبه تابع

اندیس تصویر چپ	اندیس تصویر راست
$i, j + \text{dis} - 2$	$i, j - 1$
$i, j + \text{dis} - 1$	$i, j - 1$
$i, j + \text{dis}$	$i, j - 1$

جدول (۳): اندیس های پیکسل لنگری مورد استفاده برای محاسبه تابع هزینه پیکسل لنگری انتخابی

اندیس تصویر چپ	اندیس تصویر راست
$i, j + \text{dis} - 1$	i, j
$i, j + \text{dis}$	i, j
$i, j + \text{dis} + 1$	i, j

جدول (۴): اندیس های پیکسل لنگری مورد استفاده برای محاسبه تابع هزینه پیکسل پسین پیکسل لنگری انتخابی

اندیس تصویر چپ	اندیس تصویر راست
$i, j + \text{dis}$	$i, j + 1$
$i, j + \text{dis} + 1$	$i, j + 1$
$i, j + \text{dis} + 2$	$i, j + 1$

این مشابهت ها نشان می دهد که اگر مقادیر تابع هزینه برای پیکسل های مجاور در اختیار باشد این امکان فراهم می گردد که بدون انجام محاسبات اضافه فرایند انطباق متقابل انجام پذیر باشد. این بدین معنی است که اگر در یک مرحله تنها به انطباق قطع شده و بدون انطباق متقاطع اکتفا شود، تمام داده های مورد نیاز برای انطباق متقاطع فراهم خواهد بود (جز یک ستون از پیکسل های تصاویر) و این به معنی کاهش محاسبات به رابطه‌ی (۲) خواهد بود. لازم به ذکر است تنها ستونی که داده های تابع هزینه برای انطباق متقاطع فراهم نیست، آخرین ستونی از تصویر راست است که در حالت استریوی قطع شده و بدون انطباق متقاطع، تابع

¹ Smoothness

$numOfCostCalculation = NumOfAllPixels \times KernelSize \times 3$ (۹)
 و در صورتی که انطباق متقابل فعال باشد، تعداد محاسبات برابر خواهد بود با:

$numOfCostCalculation = NumOfAllPixels \times KernelSize \times 5$ (۱۰)

حال اگر در یک مرحله تمام تابع هزینه‌ها محاسبه گردیده و فرایند استنتاج در مرحله‌ی دیگر انجام گردد آنگاه با توجه توضیحات داده‌شده در رابطه با انطباق متقابل تعداد محاسبات در پنجره انطباق برابر خواهد بود با رابطه‌ی (۱۱):

$numOfCostCalculation = NumOfAllPixels \times KernelSize \times 3$ (۱۱)

بنابراین اگر در هر پنجره استنتاج نیز محاسبه عمق همسایگی‌ها به صورت محلی بررسی گردد تعداد محاسبات برابر خواهد بود با:

$numOfCostCalculation = NumOfAllPixels \times KernelSize \times (3 + 1)$ (۱۲)

باتوجه به توضیحات داده‌شده بدون اینکه به محاسبات افزایشی قابل ذکر داشته باشد، فرض نرمی به صورت نسبی اعمال گردیده است. همان‌طور که در قسمت نتایج نمایش داده‌شده است با استفاده از روش پیشنهادی کیفیت پاسخ‌ها به مراتب بهبود پیدا کرده است.

در ادامه الگوریتمی که برای اجرا بر روی CPU مورد استفاده قرار گرفته است تشریح گردیده است و سپس نحوه‌ی پیاده‌سازی روش پیشنهادی به صورت موازی بر روی کارت گرافیک تشریح گردیده است. لازم به ذکر است با وجود افزایش اندک بار محاسباتی در این روش، باتوجه به نحوه‌ی اجرا بر روی کارت گرافیک، مدت زمان اجرا به مقدار چشمگیری کاهش پیدا کرده است (۴۳/۹۳ درصد) که قسمتی از نوآوری در این پژوهش مربوط به پیاده‌سازی روش پیشنهادی بر روی کارت گرافیک برای اجرا به صورت موازی است.

شبه‌کد (۱): شبه‌کد روش پیشنهادی برای اجرا بر روی CPU

```

1 LeftImage = LeftImage, RightImage
2 disparity = FindLayerByUltrasonicData()
3 For(i in range(Int(NCK/2), NR-Int(NCK/2))):
4     For(j in range(Int(NCK/2)+1, NC-Int(NCK/2))):
5         //3 Costs will be calculated.
6         calcCostsForAnchor(i,j,NCK,NRK,disparity,LeftImage,RightImage, Costs)
7     For(i in range(NRK,NR-NRK)):
8         For(j in range(NCK+1,NC-NCK)):
9             AcceptedSubKernel[i,j] = countAcceptedSubKernel(i,j,NCK,NRK, Costs)
10    For(i in range(NRK,NR-NRK)):
11        For(j in range(NCK+1,NC-NCK)):
12            If(AcceptedSubKernel[i,j] >= AcceptanceThreshold)
13                Result[i,j] = True
14            else
15                Result[i,j] = False;
    
```

شبه‌کد (۲): محاسبه‌ی تابع هزینه

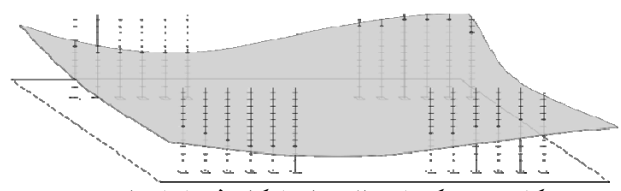
```

1 function calcCostsForAnchor(i,j,NCK,NRK,disparity,LeftImage,RightImage, Costs)
2     For(dis in range(disparity-1, disparity+1)):
3         For(v in range(-Int(NCK/2), Int(NCK/2))):
4             For(w in range(-Int(NRK/2), Int(NRK/2))):
5                 Delta = RightImage(i+v,j+w) - LeftImage(i+v,j+dis+w);
6                 Costs[i,j,k] += pow(Delta,2)
    
```

شبه‌کد (۳): فرایند استنتاج

```

1 function countAcceptedSubKernel(i,j,NCK,NRK, Costs)
2     Counter = 0
3     For(v in range(-Int(NCK/2), Int(NCK/2))):
4         For(w in range(-Int(NRK/2), Int(NRK/2))):
5             If(Costs[i+v,j+w,1] < Costs[i+v,j+w,0] & Costs[i+v,j+w,1] <
                Costs[i+v,j+w,2]):
6                 Counter = Counter+1
7     Return Counter
    
```

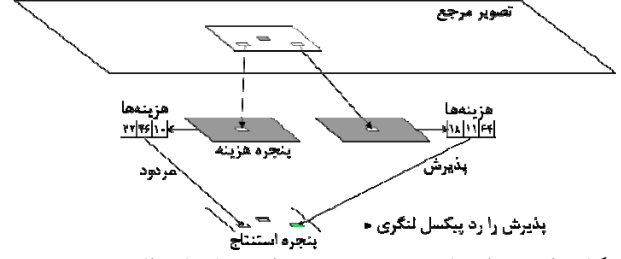


شکل (۳): یکی از حالت‌های امکان‌پذیر از ایجاد پوش

با محدود کردن محاسبات به لایه‌های پیشین و پسین لایه شناسایی شده با حسگر التراسونیک، قسمتی از پوش تولیدشده نامعتبر می‌گردد. لذا استفاده از روش استریوی جهانی بدون تغییر در کنار روش استریوی قطع‌شده قابل استفاده نیست. در همین راستا محققین پیشنهاد داده‌اند که بینایی استریوی جهانی به نحوی استفاده شود که با حفظ نسبی مزیت‌های آن، بار محاسباتی نیز افزایش چشمگیری پیدا نکند. باتوجه به توضیحات داده‌شده، محققین پیشنهاد کرده‌اند که دو نمونه پنجره در فرایند انطباق مورد استفاده قرار گیرد. این دو نوع پنجره شامل موارد زیر است.

- پنجره‌ی محاسبه‌گر تابع هزینه
- پنجره استنتاجی

در این روش به ازای تمام پیکسل‌ها به جای اینکه تابع هزینه تابعی فقط شدت نور همسایگی‌ها و یا تمام عمق‌ها پیکسل‌های موجود در همسایگی باشد، تابع هزینه محدود به مقایسه شدت نور در پنجره‌های انطباقی و البته عمق شناسایی شده در پیکسل‌های موجود در همسایگی (تنها درون پنجره انطباق) خواهد بود. شکل (۴) نمایی از روند پیشنهادی به نمایش گذاشته شده است.



شکل (۴): نمایی از روند روش پیشنهادی برای اعمال نرمی در روش بینایی استریوی قطع‌شده.

با این کار با وجود اعمال فرض نرمی، هم بار محاسباتی افزایش قابل توجهی پیدا نکرده و هم کیفیت پاسخ به نسبت روش محلی استریوی قطع‌شده، بهبود پیدا می‌کند. در این روش در یک مرحله تمام پیکسل‌ها روش بینایی استریوی قطع‌شده را طی می‌کنند و هر پیکسل لنگری هر سه مقدار تابع هزینه‌ی این گام خود را نگهداری می‌کند و در مرحله‌ی دوم که پنجره‌ی استنتاج است، درصد پذیرش پیکسل‌هایی که در همسایگی هر پیکسل لنگری است، محاسبه خواهد شد و در صورت بیشتر بودن از حد آستانه‌ی تعیینی توسط کاربر، آن پیکسل به عنوان جزئی از مانع به حساب خواهد آمد.

تعداد محاسباتی که در این روش بایستی انجام گردد برابر خواهد بود با:

همچنین تعداد بلوک‌ها در راستای z نیز به دلیل انجام فرایند انطباق در سه لایه‌ی معرفی شده است. لازم به ذکر است که باتوجه به توضیحات داده شده، برای انجام فرایند انطباق متقاطع بحث شده، نیازی به افزایش تعداد بلوک‌ها در راستای z به ۵ نیست و همین تعداد بلوک محاسباتی کفایت خواهد کرد. در هر بلوک نیز تعداد نخ‌های محاسباتی در راستای x ، y و z به ترتیب برابر با روابط (۱۶) تا (۱۸) است.

$$BlockDimX = NCK \quad (16)$$

$$BlockDimY = NRK \quad (17)$$

$$BlockDimZ = 1 \quad (18)$$

درواقع ابعاد بلوک‌های محاسباتی برابر با ابعاد پنجره انطباق و پنجره استنتاج در نظر گرفته شده است تا در یک گام به محاسبه‌ی اختلاف‌ها و در گام بعدی نیز شمارش زیر پنجره‌های معتبر در شناسایی مانع را پوشش دهد.

منظور از زیر پنجره‌های معتبر، زیر پنجره‌هایی است که تابع هزینه در لایه‌ی میانی به نسبت دو لایه‌ی دیگر کمینه بوده است. در پیاده‌سازی روش پیشنهادی بر روی کارت گرافیک، در دو سطح هم‌زمان‌سازی مورد استفاده قرار گرفته است.

- هم‌زمان‌سازی در سطح هر بلوک
- هم‌زمان‌سازی در سطح شبکه بلوک

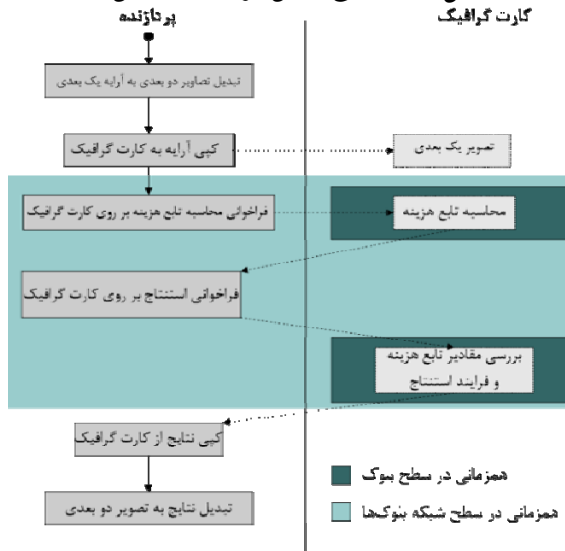
این دو نوع هم‌زمان‌سازی به ۳ منظور ایجاد گردیده است که شامل موارد زیر است.

- هم‌زمان‌سازی در هر بلوک به منظور تجمیع تابع هزینه به‌ازای هر پیکسل و یک‌میزان انحراف

- هم‌زمان‌سازی در سطح شبکه‌ی بلوک‌ها به منظور ایجاد فرصت برای تکمیل محاسبات تابع هزینه در ۳ لایه‌ی انحراف

- هم‌زمان‌سازی در هر بلوک برای شمارش زیر پنجره‌های تأییدکننده مانع (پنجره استنتاج)

در شکل (۷) نمایی از این فرایند به نمایش گذاشته شده است:



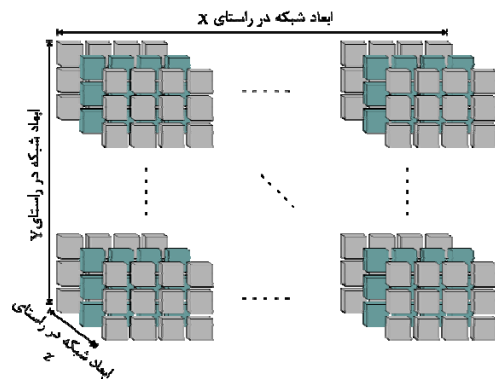
شکل (۷): سطوح هم‌زمان‌سازی در روش پیشنهادی.

به نحوی که عبارت NC بیانگر تعداد ستون‌های پیکسل و NR تعداد سطرهای پیکسل در تصویر مورد استفاده است و عبارت NCK بیانگر تعداد ستون‌های موجود در پنجره تطبیق است و NRK بیانگر تعداد سطرهای موجود در پنجره انطباق است. لازم به ذکر است که در این پژوهش ابعاد پنجره انطباق و پنجره استنتاج یکسان در نظر گرفته شده است.

۲-۳- افزایش سرعت

به منظور انجام هر چه سریع‌تر الگوریتم استریوی پیشنهادی و دستیابی به الگوریتمی زمان واقعی نحوه‌ی پیاده‌سازی روش پیشنهادی برای اجرا بر روی کارت‌های گرافیکی نیز ارائه گردیده است.

بنابراین شبکه از بلوک‌های محاسباتی به شکل ۵ زیر ایجاد گردیده است:



شکل (۵): ساختار شبکه‌ی بلوک‌های مورد استفاده.

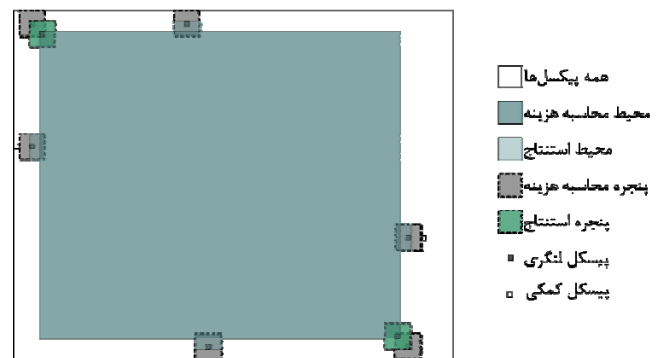
تعداد بلوک‌ها در راستای x ، y و z به ترتیب برابر خواهد بود با روابط (۱۳) تا (۱۵).

$$GridDimX = NC - (2 \times NCK) - (Disparity + 1) \quad (13)$$

$$GridDimY = NR - (2 \times NRK) \quad (14)$$

$$GridDimZ = 3 \quad (15)$$

دلیل انتخاب این ابعاد از شبکه‌ی بلوک‌ها در راستای x و y محدودی معتبر برای فرایند استنتاج است که در شکل (۶) نمایش داده شده است.



شکل (۶): محدودهای عملیاتی در هر فاز شناسایی مانع.

لذا باتوجه به توضیحات داده شده محل ذخیره‌سازی به صورت رابطه‌ی (۲۸) است.

$$\text{Costs_blockIndex} = (\text{vIndexAnchor} \times (\text{NC}) \text{uIndexAnchor}) \times 3 + \text{blockIndexZ} \quad (28)$$

پس از اطمینان از ذخیره‌سازی همه‌ی مقادیر تابع هزینه‌ی محاسبه‌شده توسط بلوک‌ها (همگام‌سازی در سطح شبکه بلوک‌ها) به انجام فرایند استنتاج پرداخته می‌شود. بنابراین روند اجرا از کارت گرافیک به CPU منتقل می‌گردد و برای ادامه تابع دیگر برای محاسبه‌ی تعداد زیر پنجره‌های پذیرفته‌شده بر روی داده‌های موجود بر روی حافظه‌ی کارت گرافیک فراخوانی خواهد شد. لازم به ذکر است که فرایند استنتاج با استفاده از همان شبکه‌ی بلوک‌ها و نخ‌های محاسباتی دنبال خواهد شد.

به منظور انجام فرایند استنتاج باتوجه به این‌که در هر صفحه از شبکه‌ی بلوک‌ها، تعداد بلوک‌ها برابر با تعداد پیکسل‌های معتبر جهت فرایند استنتاج است، از تمام بلوک‌های موجود در لایه میانی بلوک‌ها بهره برده خواهد شد، از طرفی نیز باتوجه به اینکه برای فرایند استنتاج تعداد زیر پنجره‌ها برابر با تعداد نخ‌های محاسباتی بود، در ادامه از تمام نخ‌های بلوک‌های مذکور بهره برده خواهد شد. فرایند استنتاج بدین صورت خواهد بود که در هر بلوک، هر نخ وظیفه‌ی بررسی ۳ یا ۵ تابع هزینه را بر عهده خواهد داشت. در صورتی‌که بررسی چک متقابل فعال باشد، ۵ تابع هزینه مقایسه می‌شود و در صورتی‌که چک متقابل فعال نباشد سه تابع هزینه مقایسه می‌گردد. روند مقایسه در حالتی که چک متقابل فعال نباشد تنها به سه تابع هزینه‌ی متناظر با رنگ‌های خاکستری در شکل (۹) منحصر است. اگر چک متقابل فعال باشد ۵ تابع هزینه‌ای که در شکل (۹) در کادر سبزرنگ مشخص گردیده مقایسه می‌گردند. در هر دو صورت اگر تابع هزینه‌ی میانی از تمام مورد دیگر کمتر بود، آن زیر پنجره به‌عنوان یک زیر پنجره تأییدکننده مانع در نظر گرفته می‌شود.



شکل (۸): تابع هزینه‌های مورد نیاز برای فرایند انطباق با چک متقابل فعال بر روی آرایه‌ی مقادیر تابع هزینه.

باتوجه به توضیحات داده‌شده، هر نخ در بلوک‌های میانی، ۳ یا ۵ تابع هزینه را مقایسه کرده و در صورت تأیید وجود مانع، در یک آرایه‌ی اشتراکی بین نخ‌های هر بلوک، عدد ۱ و در غیر این صورت عدد ۰ ثبت می‌شود. پس از اتمام بررسی (دومین هم‌زمانی در سطح بلوک) اولین نخ از هر بلوک به تجمیع نتایج می‌پردازد و در صورتی‌که تعداد زیر پنجره‌های تأییدکننده وجود مانع از حد آستانه تعیین شده فراتر رود، آنگاه آن پیکسل به‌عنوان جزئی از مانع در نظر گرفته می‌شود و در یک آرایه‌ی یک‌بعدی که مشابه تصاویر یک‌بعدی دریافتی است، ذخیره می‌گردد. در نهایت پاسخ از روی

به منظور پیاده‌سازی روش پیشنهادی بر روی کارت گرافیک در ابتدا تصاویر به رشته‌ای یک‌بعدی از شدت نورها تبدیل شده‌اند و به درون کارت گرافیک کپی گردیده است. سپس در اولین مرحله‌ی هم‌زمانی بلوک‌ها، هر نخ در هر بلوک وظیفه‌ی محاسبه‌ی اختلاف شدت نور دو پیکسل را بر عهده خواهد داشت. به زبان دیگر هر بلوک وظیفه‌ی به دست آوردن اختلاف نور پیکسل‌ها در پنجره انطباق و تجمیع آن به‌ازای مقدار مشخصی از انحراف را به عهده خواهند داشت. این میزان انحراف با استفاده از رابطه‌ی (۱۹) به دست خواهد آمد.

$$\text{BlockDisparity} = \text{Disparity} + \text{block.z} - 1 \quad (19)$$

موقعیت پیکسل‌های مذکور از رابطه‌ی (۲۰) تا (۲۳) به دست خواهد آمد.

$$\text{uIndexLeft} = \text{blockIndexX} + \text{threadIndexX} \quad (20)$$

$$\text{vIndexLeft} = \text{blockIndexY} + \text{threadIndexY} \quad (21)$$

$$\text{uIndexRight} = \text{uIndexLeft} + \text{BlockDisparity} \quad (22)$$

$$\text{vIndexRight} = \text{vIndexLeft} \quad (23)$$

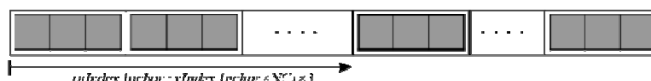
در نهایت باتوجه به رابطه‌ی (۲۴) اندیس شدت نور هر پیکسل در سطر (uIndex) و ستون (vIndex) در هر یک از تصاویر دوبعدی چپ و راست به اندیس شدت نور در شکل یک‌بعدی شده تبدیل خواهد شد.

$$1DIndex = \text{uIndex} + \text{numOfColumns} \times \text{vIndex} \quad (24)$$

پس از به دست آمدن اندیس‌های پیکسل‌های مذکور در فرم یک‌بعدی، اختلاف شدت نورها در یک آرایه‌ی یک‌بعدی اشتراکی در سطح هر بلوک ذخیره خواهد شد. این ذخیره‌سازی به صورت رابطه‌ی (۲۵) خواهد بود.

$$\begin{aligned} \text{Diff}[\text{threadIndexX} + \text{threadIndexY} \times \text{NCK}] \\ = (\text{LeftImage1D}[\text{1DIndexLeft}] \\ - \text{RightImage1D}[\text{1DIndexRight}])^2 \end{aligned} \quad (25)$$

در ادامه پس از اتمام محاسبات تمام نخ‌های محاسباتی در هر بلوک (اولین مرحله هم‌زمانی در سطح بلوک)، اولین نخ به تجمیع مقادیر موجود در آرایه‌ی اشتراکی Diff می‌پردازد و در حافظه‌ی اشتراکی بین شبکه بلوک‌ها ذخیره می‌کند. طول این آرایه برابر با $\text{NumOfAllPixels} \times 3$ است. محل ذخیره‌سازی به نحوی در نظر گرفته شده است که هر سه تابع هزینه‌ی متناظر با هر زیر پنجره انطباق به صورت متوالی پشت سرهم قرار داده شوند. در شکل- (۸) نحوه‌ی ذخیره‌سازی به نمایش گذاشته شده است.



شکل (۸): نحوه‌ی ذخیره‌سازی مقادیر تابع هزینه در حافظه‌ی اشتراکی بین بلوک‌ها.

به طوری‌که:

$$\text{uIndexAnchor} = \text{blockIndexX} + \text{int}(\text{NCK}/2) \quad (26)$$

$$\text{vIndexAnchor} = \text{blockIndexY} + \text{int}(\text{NRK}/2) \quad (27)$$

پژوهش کاهش زمان انجام فرایند شناسایی مانع علاوه بر افزایش کیفیت پاسخ‌ها بوده است. در همین راستا نتایج زمانی اجرا در دو حالت، اجرای کامل بر روی CPU و اجرا شناسایی با بهره‌گیری از GPU دنبال گردیده است.

۳- نتایج

کارت گرافیک به بیرون کپی شده و آرایه‌ی یک بعدی نتایج به تصویر دوبعدی خروجی تبدیل می‌گردد.

جدول (۵): مقایسه‌ی کمی درصد خطای روش پیشنهادی با پژوهش [۳۳] به‌ازای سری داده‌های مختلف و درصد آستانه‌ی پذیرش متفاوت.

پژوهش [۳۳]	۹۰	۶۰	۳۰	% آستانه/داده
(a)	۰/۱۶	۰/۳۸	۰/۷۱	۲/۱۲
(b)	۰/۲۹	۰/۲۲	۰/۵۶	۱/۰۷
(c)	۰/۱۹	۰/۳۸	۰/۶۵	۲/۱۰
(d)	۱/۶۹	۱/۱۴	۱/۷۱	۱/۸۶

جدول (۶): درصد خطای کاهش یافته در داده‌های مختلف به‌ازای درصد آستانه‌ی مختلف به نسبت نتایج پژوهش [۳۳].

% آستانه/داده	۹۰	۶۰	۳۰
(a)	۹۲/۴۵	۸۲/۰۸	۶۶/۵۱
(b)	۷۲/۹۰	۷۹/۴۴	۴۷/۶۶
(c)	۹۰/۹۵	۸۱/۹۰	۶۹/۰۵
(d)	۹/۱۴	۳۸/۷۱	۸/۰۶
میانگین کل	۶۱/۵۷		

از پارامترهای تأثیرگذار در مدت‌زمان اجرای روش پیشنهادی و مقایسه‌ی آن با پژوهش‌های پیشین، سخت‌افزار مورد استفاده است. لذا در این پژوهش دقیقاً از همان سخت‌افزار و داده استفاده‌شده در پژوهش [۳۳] استفاده گردیده است که مشخصه‌های فنی سخت‌افزار مذکور به شرح زیر است.

CPU: Intel Core i7, 7700k

GPU: Nvidia 1080 Ti

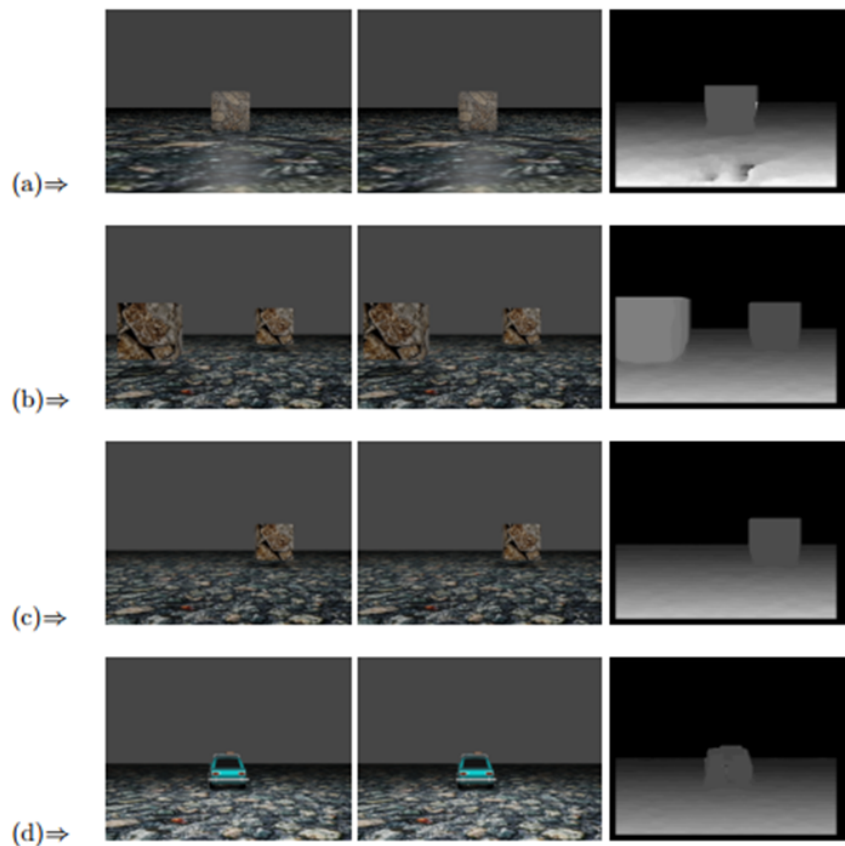
نتایج زمانی روش پیشنهادی در مقایسه با پژوهش‌های پیشین برای اجرای کامل با استفاده از CPU در جدول (۷) ارائه گردیده است. همان‌طور که از نتایج مشخص است با توجه به استفاده از پنجره استنتاج مقداری مدت‌زمان پردازش برای اجرا بر روی CPU افزایش پیدا کرده است، اما با توجه به روش پیشنهادی برای اجرا بر روی کارت گرافیک، این نقص نیز مرتفع گردیده است. در ادامه نتایج مربوط به این ارتقا نیز ارائه گردیده است.

نتایج حاصله از این پژوهش در دو زیر بخش، ارتقا کیفیت پاسخ و ارتقا سرعت پردازش ارائه و مورد بررسی قرار گرفته است. به‌منظور بررسی نتایج روش پیشنهادی از داده‌های ارائه‌شده در پژوهش [۳۳] استفاده گردیده است. در پژوهش مذکور محققان با استفاده از نرم‌افزار متن‌باز بلندر محیط شبیه‌سازی تشریح شده در قسمت الگوریتم را ایجاد کرده و از داده‌های آن در پژوهش مذکور بهره برده‌اند. این تصاویر در شکل (۱۰) ارائه گردیده است. اولین ستون از سمت چپ، تصاویر دوربین چپ، و دومین ستون، تصاویر دوربین راست، و ستون سوم حل استریوی محلی داده‌های مذکور است.

همان‌طور که در تشریح روش پیشنهادی توضیح داده شد، در مرحله‌ی استنتاج روش پیشنهادی، شرط پذیرش پیکسل لنگری به‌عنوان جزئی از مانع، تأیید شدن تعدادی پیکسل لنگری در زیر پنجره‌های انطباق است. این تعداد به‌عنوان حد آستانه‌ی پذیرش تعریف گردیده است که اگر به تعداد کل پیکسل‌ها موجود در پنجره‌ی انطباق تقسیم گردد، درصد آستانه‌ی پذیرش حاصل خواهد شد. در این راستا نتایج حاصله از این روش در چند درصد آستانه‌ی پذیرش مختلف تولید گردیده است. در شکل (۱۱) در سه ستون اول از سمت چپ، نتایج حاصله از روش پیشنهادی به‌ازای درصد آستانه‌ی پذیرش ۳۰، ۶۰ و ۹۰ درصد و به‌ازای سری داده‌های مختلف قرار داده‌شده است. همان‌طور که به‌صورت کیفی نیز مشهود است، کیفیت پاسخ‌های روش پیشنهادی به نسبت پژوهش [۳۳] (ستون چهارم شکل ۱۱) ارتقا پیدا کرده است. اهمیت این ارتقا زمانی مشخص می‌شود که علاوه بر افزایش کیفیت پاسخ، مدت‌زمان اجرای روش برای پیدا کردن مانع نیز ۴۳/۹۳ درصد کاهش پیدا کرده است. به‌منظور نمایش کمی ارتقای صورت گرفته در روش پیشنهادی، جدول (۵) و (۶) ارائه گردیده است.

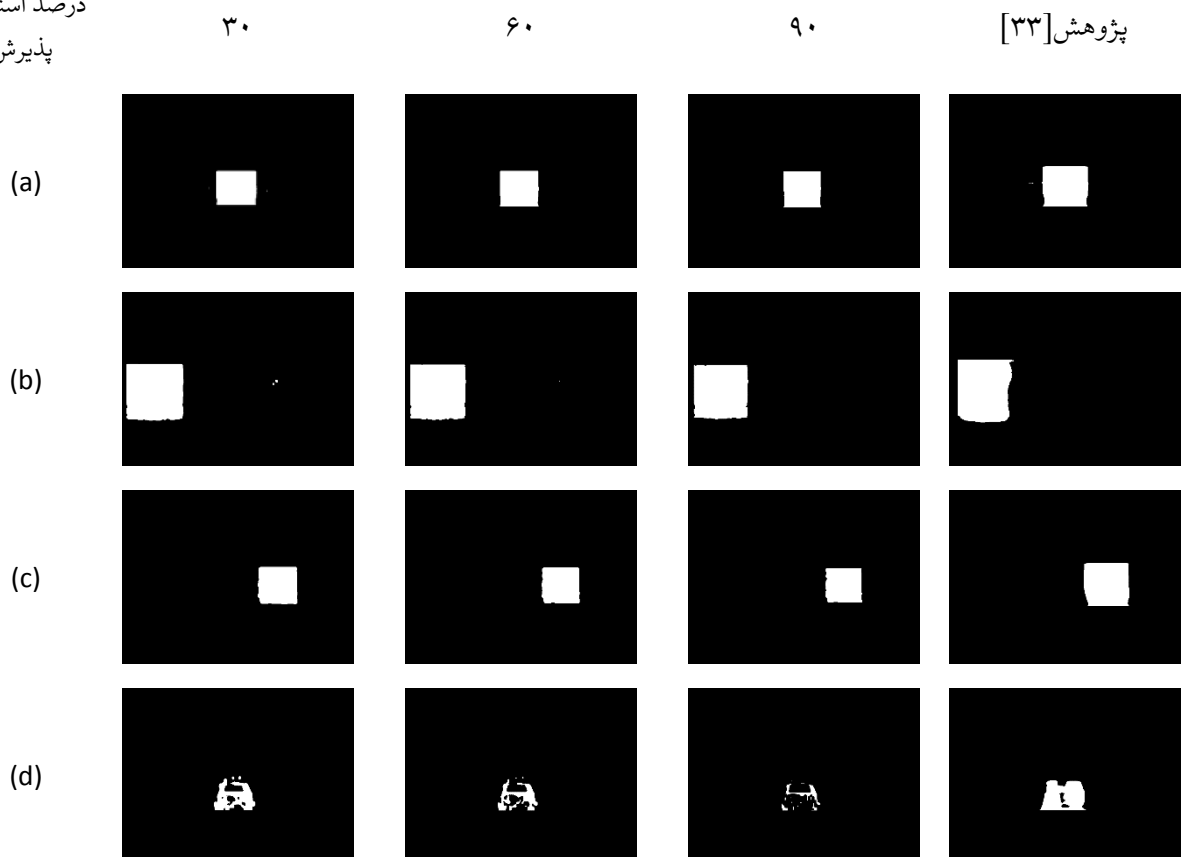
همان‌طور که نتایج ارائه‌شده در جدول (۶) نشان می‌دهند، روش پیشنهادی کیفیت پاسخ به میزان چشم‌گیری بهبود بخشیده است (۶۱/۵۷ درصد). لازم به ذکر است که در پژوهش [۳۳] با توجه به نقص‌هایی که در روش‌ها وجود داشت، سبب اجبار محققین به استفاده از فیلترهای پس پردازش می‌شد، اما در روش پیشنهادی بدون استفاده از فیلتر نتایج درصد خطا به‌شدت کاهش پیدا کرده است. در ادامه به ارائه‌ی مدت‌زمان مورد نیاز برای اجرای روش پیشنهادی با استفاده از CPU و GPU پرداخته شده است.

همان‌طور که در قسمت مقدمه توضیح داده شد، یکی از مهم‌ترین عوامل محدودکننده در استفاده از روش‌های شناسایی مانع مبتنی بر پردازش تصویر، بار محاسباتی بالا و مدت‌زمان اجرای روش‌های شناسایی مانع است. لذا یکی از مهم‌ترین رکن‌ها این



شکل (۱۰): داده‌های مورد استفاده در پژوهش [۳۳] و پژوهش حاضر به منظور ارزیابی روش پیشنهادی.

درصد آستانه‌ی پذیرش



شکل (۱۱): نتایج روش پیشنهادی به‌ازای درصد آستانه‌ی پذیرش مختلف و سری داده‌های ارائه‌شده در پژوهش [۳۳] در مقایسه با نتایج آن.

جدول (۷): مدت زمان اجرای روش پیشنهادی بر روی CPU و مقایسه آن با نتایج [۳۳] (زمان‌ها برحسب میلی ثانیه است).

داده	(a)			(b)			(c)			(d)		
	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰
% آستانه	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰
زمان اجرا با CPU	۱۲۹۸	۱۲۶۳	۱۲۷۸	۱۲۶۱	۱۲۷۱	۱۲۵۸	۱۲۸۰	۱۲۷۶	۱۲۵۲	۱۲۸۲	۱۲۹۳	۱۲۴۰
میانگین زمان اجرا با CPU		۱۲۸۰			۱۲۶۳			۱۲۷۰			۱۲۷۲	
پژوهش [۳۳]		۱۱۵۳			۱۱۵۳			۱۱۶۰			۱۱۵۴	

جدول (۸): مدت زمان اجرای گام‌های مختلف روش پیشنهادی به ازای سری داده‌های مختلف و درصد آستانه پذیرش متفاوت بر روی GPU (زمان‌ها برحسب میلی ثانیه است).

داده	(a)			(b)			(c)			(d)		
	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰
% آستانه/فعالیت	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰	۳۰	۶۰	۹۰
خواندن تصاویر	۸/۸۵	۸/۶۵	۸/۷۰	۸/۶۷	۹/۴۵	۹/۲۶	۹/۱۴	۸/۸۵	۸/۷۰	۹/۶۰	۸/۷۴	۸/۸۳
تبدیل به آرایه 1D	۱/۰۲	۱/۰۳	۱/۰۰	۱/۰۰	۱/۰۳	۱/۱۷	۱/۰۱	۰/۹۹	۱/۲۴	۱/۱۱	۱/۰۲	۱/۳۰
کپی به کارت گرافیک	۰/۱۶	۰/۲۱	۰/۱۴	۰/۱۴	۰/۱۴	۰/۱۵	۰/۱۴	۰/۱۳	۰/۳۰	۰/۱۴	۰/۱۴	۰/۳۱
انجام محاسبه با کارت گرافیک	۰/۰۴	۰/۰۴	۰/۰۳	۰/۰۴	۰/۰۳	۰/۰۳	۰/۰۴	۰/۰۳	۰/۰۷	۰/۰۳	۰/۰۳	۰/۰۶
کپی نتایج از کارت گرافیک	۸/۸۴	۸/۹۰	۸/۸۷	۸/۷۸	۸/۹۲	۸/۹۳	۸/۷۶	۸/۷۷	۸/۷۱	۸/۷۷	۸/۷۶	۸/۷۴
تولید خروجی	۱/۱۴	۱/۱۳	۱/۱۳	۱/۱۳	۱/۳۰	۱/۱۴	۱/۱۳	۱/۱۳	۱/۱۴	۱/۱۳	۱/۱۳	۱/۱۴
مجموع	۲۰/۰۵	۱۹/۹۷	۱۹/۸۸	۱۹/۷۶	۲۰/۸۶	۲۰/۶۷	۲۰/۲۲	۱۹/۹۱	۲۰/۱۶	۲۰/۷۸	۱۹/۸۲	۲۰/۳۸
میانگین		۱۹/۹۷			۲۰/۴۳			۲۰/۱۰			۲۰/۳۲	
میانگین کلی							۲۰/۲۱					

۴- نتیجه‌گیری

در این پژوهش محققین به دنبال افزایش سرعت و دقت شناسایی مانع با استفاده از ترکیب داده‌های استریو و التراسونیک بوده‌اند. در این پژوهش، روشی که با استفاده از ترکیب داده‌های استریو و التراسونیک وجود داشت در دو حوزه دقت و سرعت شناسایی مانع بهبود داده شده است. در این پژوهش محققین روشی را پیشنهاد داده‌اند که به کمک آن بتوان علاوه بر شباهت شدت نور در پنجره‌های انطباق، از موقعیت عمق همسایگی‌ها به صورت محدود شده و کافی استفاده شود. محققین در این پژوهش موفق شده‌اند که با محدودسازی ارائه شده در استفاده از عمق پیکسل‌ها موجود در همسایگی، بدون آنکه افزایش چشمگیری در محاسبات به وجود آورند، میزان خطای روش پیشین را به مقدار چشمگیری کاهش دهند (خطاها به میزان ۶۱/۵۷ درصد کاهش پیدا کرده است) در ادامه با توجه به نحوه پیاده‌سازی که برای اجرای روش پیشنهادی بر روی GPU ارائه گردیده است، مدت زمان شناسایی مانع ۴۳/۹۳ درصد بهبود داده شده است. این ارتقا سبب شده است که شناسایی مانع از ۲۸ فریم بر ثانیه به ۴۹ فریم بر ثانیه افزایش پیدا کند و این بدین معنی است که اگر خودرو نیاز به شناسایی محیط در هر یک متر حرکت داشته باشد، بتواند با سرعت ۱۷۸/۱ km/h حرکت کند. بهبود هم‌زمان دقت و سرعت شناسایی مانع سبب می‌شود که خودروهای هوشمند بتوانند با سرعت بیشتری حرکت کنند و در صورت ثابت بودن سرعت، قابلیت اعتماد بیشتری داشته باشند.

نتایج حاصله از اجرای روش پیشنهادی با استفاده از GPU به صورت جدول (۸) است. در این جدول مدت زمان صرف شده به ازای گام‌های مختلف روش پیشنهادی برحسب میلی ثانیه و به ازای سری داده‌های مختلف گزارش گردیده است. همان‌طور که از جدول مذکور نتیجه‌گیری می‌شود، میانگین مدت زمان اجرای روش پیشنهادی برابر با ۲۰/۲۱ میلی ثانیه است.

به منظور ارائه مقایسه‌ی مدت زمان اجرای روش پیشنهادی با پژوهش‌های پیشین جدول (۹) گزارش گردیده است. همان‌طور که نشان داده شده است، روش پیشنهادی سبب بهبود ۴۳/۹۳ درصدی زمان شناسایی مانع گردیده است. این بدان معنی است که اگر به ازای هر یک متر حرکت خودرو به جلو به یک فریم شناسایی مانع نیاز باشد، خودرو می‌تواند با سرعت ۱۷۸/۱ km/h حرکت کند، این در صورتی است که در پژوهش پیشین، این سرعت برابر با ۱۰۰/۸ km/h بوده است.

جدول (۹): مقایسه میانگین زمانی (برحسب میلی ثانیه) اجرای پژوهش فعلی با پژوهش [۳۳]

داده	(a)	(b)	(c)	(d)	میانگین
پژوهش حاضر	۱۹/۹۷	۲۰/۴۳	۲۰/۱۰	۲۰/۳۲	۲۰/۲۱
پژوهش [۳۳]	۳۶/۰۴	۳۶/۶۱	۳۶/۰۹	۳۵/۴۲	۰۴/۳۶
درصد بهبود	۴۴/۵۹	۴۲/۶۲	۴۴/۳۱	۴۴/۱۹	۴۳/۹۳

مراجع

- matching*. Journal of Real-Time Image Processing, 2020. 17(5): p. 1447-1468.
- [16] Hirschmuller, H. *Stereo vision in structured environments by consistent semi-global matching*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). 2006. IEEE.
- [17] Michael, M., et al. *Real-time stereo vision: Optimizing semi-global matching*. in 2013 IEEE Intelligent Vehicles Symposium (IV). 2013. IEEE.
- [18] Lu, B., et al., *An improved graph cut algorithm in stereo matching*. Displays, 2021. 69: p. 102052.
- [19] Hong, L. and G. Chen. *Segment-based stereo matching using graph cuts*. in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. 2004. IEEE.
- [20] Portugal, D., et al., *An autonomous all terrain robotic system for field demining missions*. Robotics and Autonomous Systems, 2015. 70(C): p. 126-144.
- [21] Song, T., et al., *An accurate 3-D fire location method based on sub-pixel edge detection and non-parametric stereo matching*. Measurement, 2014. 50: p. 160-171.
- [22] Kneip, J., P. Fleischmann, and K. Berns, *Crop edge detection based on stereo vision*. Robotics and Autonomous Systems, 2020. 123: p. 103323.
- [23] Zbontar, J. and Y. LeCun. *Computing the stereo matching cost with a convolutional neural network*. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [24] Hamid, M.S., et al., *Stereo matching algorithm based on deep learning: A survey*. Journal of King Saud University-Computer and Information Sciences, 2020.
- [25] Dollar, P., et al., *Pedestrian detection: An evaluation of the state of the art*. IEEE transactions on pattern analysis and machine intelligence, 2011. 34(4): p. 743-761.
- [26] Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. IEEE computer society conference on computer vision and pattern recognition (CVPR'05). 2005. Ieee.
- [27] Hinton, G.E., *Learning multiple layers of representation*. Trends in cognitive sciences, 2007. 11(10): p. 428-434.
- [28] Bengio, Y., *Learning deep architectures for AI*. Foundations and trends. Machine Learning, 2009. 2(1): p. 1-127.
- [29] Ramos, S., et al. *Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling*. IEEE Intelligent Vehicles Symposium (IV). 2017. IEEE.
- [30] Zuo, X., et al. *Lic-fusion: Lidar-inertial-camera odometry*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019. IEEE.
- [1] Wang, H., P. Hu, and H. Wang, *A genetic timing scheduling model for urban traffic signal control*. Information Sciences, 2021. 576: p. 475-483.
- [2] Kendoul, F., *Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems*. Journal of Field Robotics, 2012. 29(2): p. 315-378.
- [3] Wang, Z., *Review of real-time three-dimensional shape measurement techniques*. Measurement, 2020. 156: p. 107624.
- [4] Cao, M., et al., *Fast incremental structure from motion based on parallel bundle adjustment*. Journal of Real-Time Image Processing, 2021. 18(2): p. 379-392.
- [5] Kim, W.-S., et al., *Stereo-vision-based crop height estimation for agricultural robots*. Computers and Electronics in Agriculture, 2021. 181: p. 105937.
- [6] Wu, B., et al., *Closed-Loop Pose Control and Automated Suturing of Continuum Surgical Manipulators With Customized Wrist Markers Under Stereo Vision*. IEEE Robotics and Automation Letters, 2021. 6(4): p. 7137-7144.
- [7] Han, C., et al. *Adaptive Weight Based Sparse Block Aggregation Algorithm for Stereo Matching*. in 2019 9th International Conference on Information Science and Technology (ICIST). 2019. IEEE.
- [8] Nefti-Meziani, S., et al., *3D perception from binocular vision for a low cost humanoid robot NAO*. Robotics and autonomous systems, 2015. 68: p. 129-139.
- [9] Ling, X., et al., *Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision*. Robotics and Autonomous Systems, 2019. 114: p. 134-143.
- [10] Liu, B., et al., *Adaptive Aggregate Stereo Matching Network with Depth Map Super-Resolution*. Sensors, 2022. 22(12): p. 4548.
- [11] Han, Y., et al., *Stereo dense image matching by adaptive fusion of multiple-window matching results*. Remote Sensing, 2020. 12(19): p. 3138.
- [12] Wang, Y., et al. *Photometric Stereo Based on Multiple Kernel Learning*. International Conference on Image and Graphics. 2021. Springer.
- [13] Dai, H., et al., *Adaptive disparity candidates prediction network for efficient real-time stereo matching*. IEEE Transactions on Circuits and Systems for Video Technology, 2021. 32(5): p. 3099-3110.
- [14] Lu, Z., et al., *A resource-efficient pipelined architecture for real-time semi-global stereo matching*. IEEE Transactions on Circuits and Systems for Video Technology, 2021. 32(2): p. 660-673.
- [15] Cambuim, L.F., et al., *An FPGA-based real-time occlusion robust stereo vision system using semi-global*

- [31] Zhang, J., M. Kaess, and S. Singh, *A real-time method for depth enhanced visual odometry*. Autonomous Robots, 2017. 41: p. 31-43.
- [32] Graeter, J., A. Wilczynski, and M. Lauer. *Limo: Lidar-monocular visual odometry*. IEEE/RSJ international conference on intelligent robots and systems (IROS). 2018. IEEE.
- [33] Gholami, F., E. Khanmirza, and M. Riahi, *Real-time obstacle detection by stereo vision and ultrasonic data fusion*. Measurement, 2022: p. 110718.



فرشاد غلامی در سال ۱۳۹۳ مدرک کارشناسی خود در رشته مهندسی مکانیک خود را از دانشگاه یاسوج دریافت نمود و در سال ۱۳۹۵ مدرک کارشناسی ارشد خود در رشته مهندسی مکانیک از دانشگاه علم و صنعت ایران اخذ نمود. وی در حال حاضر دانشجوی دکتری مهندسی مکانیک در دانشگاه علم و صنعت ایران می باشد. زمینه های تحقیقاتی مورد علاقه ایشان، بینایی ماشین، پردازش تصویر و کنترل خودروهای هوشمند می باشد.



اسماعیل خان میرزا دانشیار دانشکده مهندسی مکانیک دانشگاه علم و صنعت ایران، در سال ۱۳۹۱ دکتری مهندسی مکانیک را از دانشگاه تهران اخذ و تا به حال به عنوان عضو هیات علمی دانشکده مهندسی مکانیک، مشغول به فعالیت می باشند. زمینه تحقیقاتی مورد علاقه ایشان سیستم های کنترل هابیرید، سیستم های کنترل هوشمند، طراحی مسیر حرکت و سیستم های کنترل شبکه ای می باشد.



محمد ریاحی استاد دانشکده مهندسی مکانیک دانشگاه علم و صنعت ایران، مقاطع کارشناسی و کارشناسی ارشد خود در رشته مهندسی مکانیک را در دانشگاه ایلینویز طی نموده و پس از فارغ التحصیلی در مقطع دکتری مهندسی مکانیک از دانشگاه آیووا در آمریکا به عنوان عضو هیات علمی دانشکده مهندسی مکانیک مشغول به فعالیت می باشند. زمینه تحقیقاتی مورد علاقه ایشان پایش پایداری سیستم های مکانیکی و عیب یابی و تست های غیر تخریبی می باشد.