



## ارائه روش‌های حذف خطای لبه در عمق سنجی حاصل از حسگر کینکت به منظور بهبود مدل ابرنقاط محصولات گلخانه‌ای

امید دوستی ایرانی<sup>۱</sup>، محمود رضا گلزاریان<sup>۲\*</sup>، محمدحسین آق‌خانی<sup>۳</sup>، امیرحسین نایبی<sup>۴</sup>

۱. دانشجوی دکتری، گروه مهندسی بیوسیستم، دانشگاه فردوسی مشهد (omiddoostiirani@gmail.com)
۲. دانشیار، گروه مهندسی بیوسیستم، دانشگاه فردوسی مشهد (m.golzarian@um.ac.ir)
۳. استاد، گروه مهندسی بیوسیستم، دانشگاه فردوسی مشهد (aghkhani@um.ac.ir)
۴. استادیار، دانشکده مهندسی برق و رباتیک، دانشگاه صنعتی شاهرود (am\_nayebi@yahoo.com)

### چکیده

امروزه مدل‌سازی سه‌بعدی از گیاهان و محصولات کشاورزی کاربردهای گسترده‌ای در این حوزه پیدا کرده است. یکی از روش‌های ارزان‌قیمت و سریع در تهیه نقشه‌های عمق استفاده از حسگرهای کینکت است. یکی از معایب حسگر کینکت وجود نویز در نقشه عمق است که سبب کاهش کیفیت پردازش‌های بعدی بر روی این تصاویر می‌شود. در این پژوهش از چهار الگوریتم مبتنی بر متدهای میانگین، میانه، برازش منحنی و نزدیک‌ترین همسایه صحیح به منظور حذف نقاط نویز استفاده شده است، که روش‌های برازش منحنی و نزدیک‌ترین همسایه صحیح روش‌های پیشنهادی این پژوهش هستند. نتایج حاصل از اعمال این چهار الگوریتم نشان داد که همه الگوریتم‌ها قادر به حذف نقاط نویز از تصاویر عمق هستند. با توجه به اینکه عمق جایگزین شده به جای مقدار صفر در نقاط نویزی از اهمیت بالایی برخوردار است، بهترین و قابل‌اعتمادترین مقادیر عمق با استفاده از روش نزدیک‌ترین همسایه صحیح به دست آمده است.

**کلمات کلیدی:** نقشه عمق؛ حسگر کینکت، نقاط پروازی؛ عمق سنجی، خطای لبه

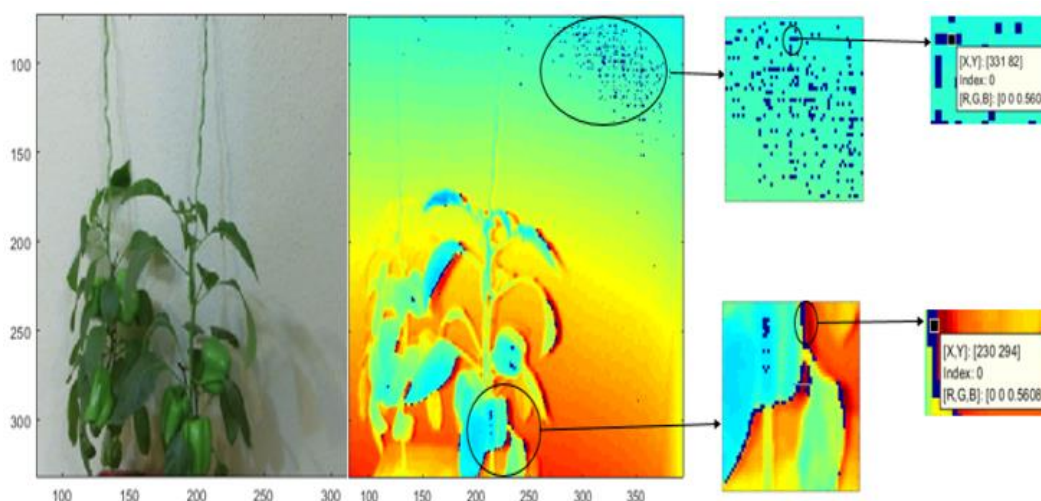
\*نویسنده مسئول: m.golzarian@um.ac.ir

## ارائه روش‌های حذف خطای لبه در عمق سنجی حاصل از حسگر کینکت به منظور بهبود مدل ابرنقاط محصولات گلخانه‌ای

### مقدمه

تعریف اصلی از نویز سیگنال‌های ناخواسته است. نویز سنسور کینکت اختلاف بین عمق اندازه‌گیری شده و مقدار واقعی است. محققین مختلف تعاریفی برای نویز ارائه کرده‌اند. از دیدگاه [1] نویز به معنای سیگنال‌های ناخواسته است. در تحقیق [2]، بیان شده است که نویز حاصل از سنسور کینکت به معنای مقادیر حاصل شده بین مقادیر واقعی و مقدار ثبت شده است. به تعبیری دیگر این نقاط پروازی<sup>۱</sup> نیز نام‌گذاری شده‌اند [4]. نقاط پروازی یکی از معضلات حسگرهای کینکت V2 است. نقاط پروازی نقاطی هستند که در تخمین عمق اشتباه تخمین زده شده‌اند و می‌توانند مربوط به لبه‌های موجود در تصویر شی و تصویر پس‌زمینه باشند. بنابراین می‌توانیم بیان کنیم که یک پیکسل  $A(x,y)$  به‌عنوان نویز شناخته و تعریف می‌شود هنگامی که مقدار معنی‌داری برای آن وجود نداشته باشد و در نقشه عمق یا تعریف نشده است یا مقدار ماکزیمم به آن نسبت داده شده است. این در حالی است که موقعیت این پیکسل به درستی تشخیص داده شده است.

کینکت یک دستگاه مرکب است که به‌عنوان یک حسگر عمق، RGB و مادون قرمز IR و صوت کار می‌کند. بنابراین حسگر کینکت 4 خروجی دارد که شامل تصویر مادون قرمز، تصویر RGB، تصویر عمق و سیگنال صوتی است [2]. در شکل (1) نمونه‌ای از تصویر RGB و تصویر نویز دار عمق مربوط به حسگر کینکت نمایش داده شده است. همان‌طور که مشاهده می‌شود این نقاط نویز می‌توانند در لبه‌های تصویر و پس‌زمینه تصویر مشاهده می‌شوند. در بررسی نقشه عمق مشاهده می‌شود که این پیکسل‌ها مقدار عددی صفر را به خود گرفته‌اند (شکل 1). وجود این نقاط نویز علاوه بر این که شکل لبه‌های تصویر را از حالت طبیعی خود خارج می‌کند، موجب کاهش دقت محاسبات پردازشی بر روی تصاویر عمق می‌گردد.



شکل 1. نمونه‌ای از تصاویر نویز دار حاصل از حسگر کینکت V2. سمت چپ، تصویر RGB. سمت راست نقشه عمق و بزرگنمایی لبه‌ها در تصویر عمق (منبع: یافته‌های تحقیق)

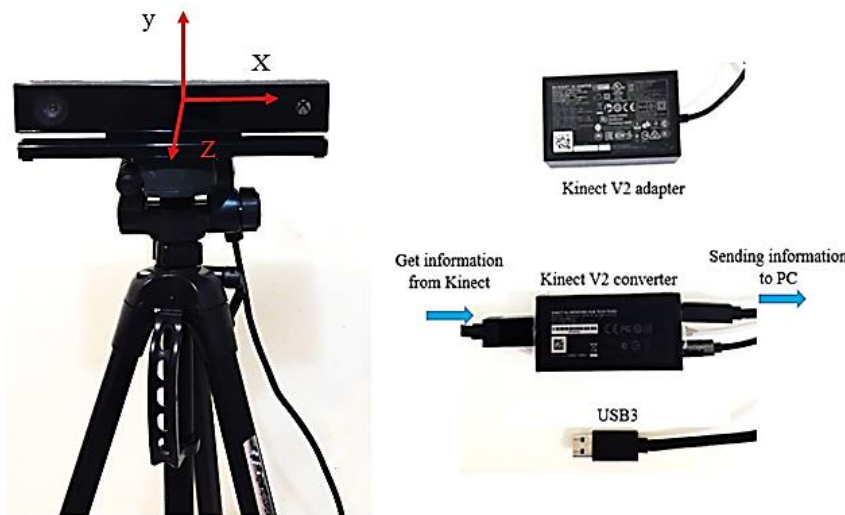
1 - Unwanted Signal  
2 Flying Pixel

با توجه به توضیحات بیان شده هدف از انجام این پژوهش ارائه روش‌های سریع به منظور حذف و جایگزینی نقاط نویز با مقادیر صحیح و مناسب است. برای این منظور در این پژوهش 4 الگوریتم مبتنی بر روش‌های آماری میانگین و میان، روش برازش منحنی و نزدیک‌ترین همسایه صحیح ارائه شده است. که در بخش‌های بعدی معرفی و مورد بررسی و ارزیابی قرار گرفته‌اند.

## مواد و روش‌ها

### دریافت تصاویر عمق

خروجی‌های حسگر Kinect V2 شامل تصویر عمق با ابعاد 512\*424 پیکسل، تصویر رنگی با ابعاد 1920\*1080 پیکسل، و تصویر مادون قرمز (IR) است. تجهیزات لازم به منظور تصویربرداری عمق در شکل (2) نمایش داده شده است که عبارت‌اند از: حسگر کینکت، پایه تصویربرداری، آداپتور، مبدل کینکت و کابل USB3.



شکل 2. نمایش محورهای مختصات بر روی کینکت (سمت چپ تصویر). آداپتور، مبدل و کابل USB جهت انتقال اطلاعات از کینکت به سیستم کامپیوتری (سمت راست تصویر).

## حذف نویز

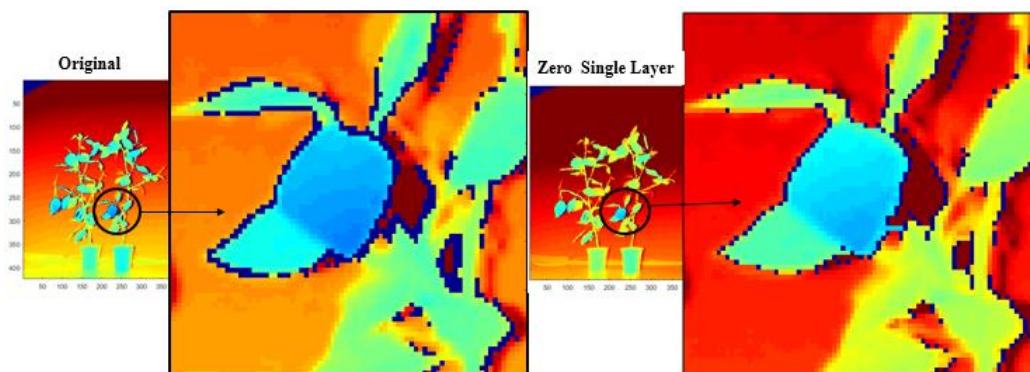
### تبدیل پیکسل‌های صفر متوالی به یک پیکسل صفر

در نقشه‌های عمق ممکن است در قسمتی از تصویر چندین لایه صفر در کنار هم وجود داشته باشد (شکل 3). در بررسی‌های انجام شده مشخص شده در قسمت‌هایی از نقشه عمق که چندین پیکسل صفر متوالی در کنار هم قرار گرفته‌اند، از کارایی لازم برخوردار نبوده و یا در صورت حذف بخشی از نقاط صفر یا همه نقاط از دقت کافی برخوردار نیستند. بنابراین در ابتدا باید تعداد لایه‌های صفر را حذف و با مقادیری صحیح جایگزین شود. به منظور حذف لایه‌های صفر از الگوریتم زیر استفاده شد نحوه انجام این الگوریتم به این صورت است.

اگر هر پیکسل از تصویر را به صورت  $A(i,j)$  نمایش دهیم، در بررسی مقادیر هر پیکسل در صورتی که در سطر  $A_m$  و ستون  $A_n$  به یک پیکسل با مقدار صفر برسیم، مقدار پیکسل بعد از آن نیز مورد بررسی قرار می‌گیرد، چنانچه مقدار پیکسل بعدی نیز صفر باشد مقدار  $A(i,j)$  با مقدار قبل از آن جایگزین می‌شود (رابطه 3). این الگوریتم را می‌توان به تعداد معینی تکرار نمود تا در همه جای نقشه عمق در صورت وجود نویز تنها یک پیکسل با مقدار صفر وجود داشته باشد.

$$\begin{cases} \text{if } A(i, j) = 0 \\ \text{and} \\ A(i, j + 1) = 0 \end{cases} \quad A(i, j) = A(i, j - 1) \quad \text{رابطه (3)}$$

پس از حذف پیکسل‌های متوالی صفر و تبدیل آن به یک پیکسل صفر (شکل 3) استفاده از الگوریتم‌های مورد استفاده در این تحقیق امکان‌پذیر می‌شود.



شکل 3. حذف لایه‌های صفر متوالی در لبه‌های اجزای گیاه

پس از حذف پیکسل‌های متوالی صفر و تبدیل آن به یک پیکسل در لبه‌های اجزای مختلف گیاه اعمال الگوریتم‌های حذف نویز با دقت و صحت بیشتری قابل انجام است. چراکه انتخاب نقاط پس‌زمینه به جای برگ یا میوه به دلیل اختلاف عمق سبب وجود خطا می‌شود. در این پژوهش به منظور انتخاب بهترین نقاط همسایگی به جای نقاط صفر از الگوریتم‌هایی به شرح زیر استفاده و در نهایت بهترین روش انتخاب شده است که در ادامه به آن پرداخته‌ایم.

### حذف نویز با استفاده از روش‌های آماری میانه و میانگین

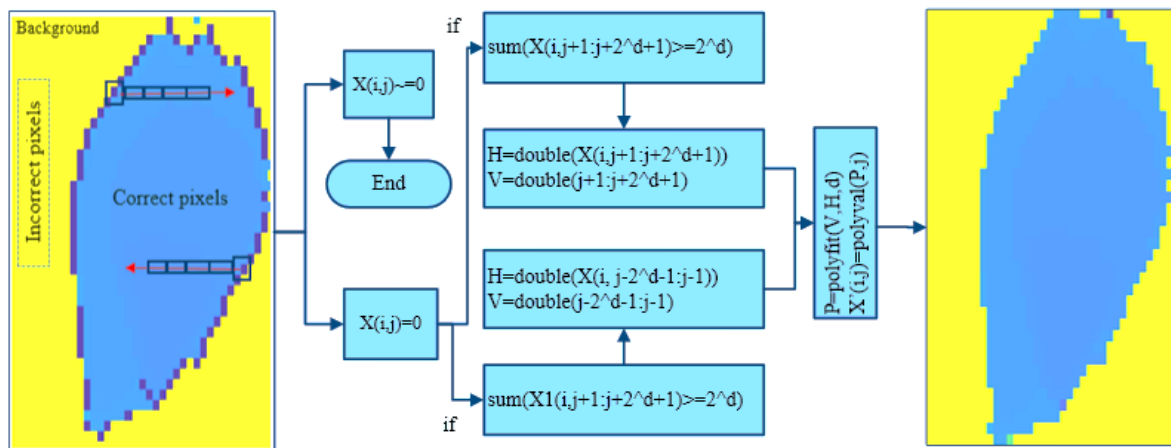
در این تحقیق به منظور حذف نویز در ابتدا از دور روش آماری میانگین و میانه در تصاویر عمق استفاده شده است. در هر دو روش ابتدا پس از فراخوانی تصویر عمق به ابعاد  $512 * 424$  یک ماسک مربعی که اندازه آن می‌تواند شامل مقادیر  $C = \{3, 4, 5, 6, \dots\}$  باشد انتخاب شده است. این ماسک با حرکت بر روی تصویر مقادیر زیر ماسک را بررسی کرده و در صورت وجود نویز این مقادیر را با مقدار میانگین یا میانه در ماسک جایگزین می‌کند. در این دو روش الگوریتم مورد استفاده در بررسی مقادیر با اعداد برابر با صفر یا غیر از صفر مواجه است. که در صورت مواجه با عدد صفر الگوریتم از مقادیر جدید به جای آن استفاده می‌کند. این دو روش در مواردی کارا بوده و در بعضی موارد کارایی مناسبی ندارند که در قسمت بحث و نتایج تفسیر شده است.

### حذف نویز با استفاده از روش برازش منحنی (Curve fitting)

با استفاده از روش برازش منحنی پس از حذف لایه‌های صفر و رسیدن به آخرین لایه صفر، تخمین مقادیر جدید به جای نقاط صفر با استفاده از تابع درجه دوم برازش شد انجام می‌شود. فرم کلی معادله برازش یک منحنی به صورت رابطه (4) است.

$$P(X) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1} \quad \text{رابطه 4}$$

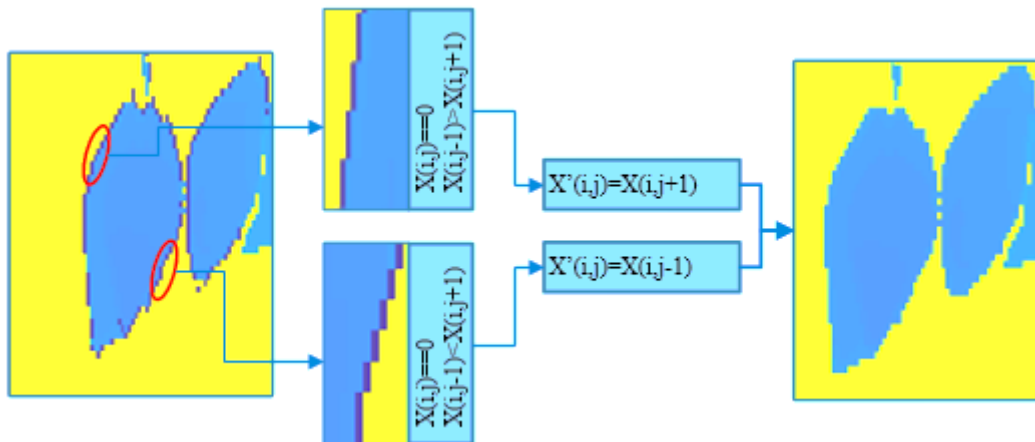
در رابطه 4 مقادیر  $p$  ضرایب معادله درجه  $n$  هستند که در الگوریتم استفاده شده در نرم‌افزار متلب با استفاده از تابع  $\text{polyfit}$  تخمین زده می‌شوند. خروجی این تابع مقادیر برازش شده برای ضرایب معادله است. در نهایت به منظور محاسبه مقادیر برازش شده از تابع  $\text{polyval}$  استفاده می‌شود. درجه تابع بر روی مقادیر تخمین زده شده تأثیر گذار است که در قسمت بحث و نتایج به آن پرداخته خواهد شد. به منظور تخمین نقاط صفر با استفاده از تابع به دست آمده بر روی نقاط همسایگی از سه همسایگی (سمت راست، سمت چپ و دو طرف) برای نقطه صفر می‌توان استفاده کرد. ولی این انتخاب این همسایگی باید به نحوی باشد که دقیق‌ترین تخمین به دست آید. به عنوان مثال هنگامی که پیکسل  $X(i,j)$  در لبه یک برگ قرار گرفته باشد نقاطی برای تخمین این نقطه می‌تواند مناسب باشد که در همسایگی آن و از داخل برگ انتخاب شوند. چراکه نقاط خارج از برگ ممکن است مربوط به سایر اجزای گیاه یا پس‌زمینه باشند که در این صورت دچار خطای بزرگی شده‌ایم. بنابراین تعداد نقاط قبل یا بعد از پیکسل با مقدار صفر به منظور فیت کردن منحنی درجه  $n$  و تخمین نقاط صفر در الگوریتم قابل تعیین است. جزئیات این الگوریتم در شکل 4 نمایش داده شده است.



شکل 4. مراحل اجرای الگوریتم برازش منحنی

#### حذف نویز متناسب با نقاط داخلی هدف (نزدیک‌ترین نقاط صحیح)

در این الگوریتم که در این پژوهش نزدیک‌ترین نقاط صحیح نامیده شده است، پس از حذف لایه‌های صفر و نزدیک شدن به نقاط لبه‌ای نقاط صفر بر اساس مقایسه عمق نقاط قبل و بعد از خود جایگزین می‌شوند. در صورتی که عمق نقطه قبل از نقطه صفر بزرگ‌تر از عمق نقطه بعد از آن باشد نقطه بعد انتخاب می‌شود و در صورتی که عمق نقطه قبل از نقطه صفر کوچک‌تر عمق نقطه بعدی باشد، عمق نقطه قبل جایگزین می‌شود (شکل 5). در این حالت از انتخاب نقاط بافاصله زیاد از نقاط مورد بررسی که منجر به ایجاد خطا می‌گردد، جلوگیری می‌شود. منظور از نقاط بافاصله زیاد نقاط در پس‌زمینه پیکسل‌های هدف است که طبیعتاً عمق بیشتری را به خود اختصاص می‌دهند. در تشخیص یک هدف در یک تصویر عمق لبه‌های تصویر هدف از اهمیت خاصی برخوردار است چراکه نقاط مرزی یا نقاط لبه بیان‌کننده شکل هندسی هدف می‌باشند. بنابراین از انتخاب نقاط پس‌زمینه به جای نقاط لبه‌ای هدف منجر به خطاهای بزرگی در محاسبات بعدی می‌گردد جلوگیری می‌شود. در شکل 5 مراحل انجام این الگوریتم نمایش داده شده است.



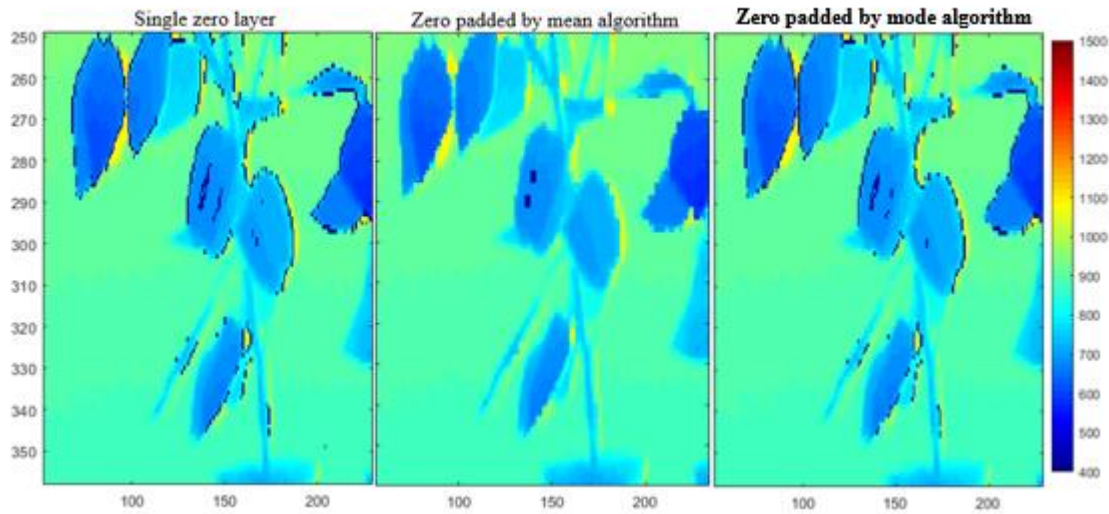
شکل 5. الگوریتم استفاده از نزدیک‌ترین نقاط صحیح

### نتیجه‌گیری

در این قسمت به بررسی نتایج حاصل از اعمال الگوریتم‌های ارائه‌شده در این تحقیق پرداخته‌شده است. همان‌طور که در بخش‌های قبلی نیز بیان‌شده است، قبل از اعمال هر الگوریتمی تعداد پیکسل‌های صفر متوالی به یک پیکسل صفر کاهش یافته است. در ادامه ابتدا نتایج حاصل از روش میانگین و فراوانی و پس از آن روش‌های برازش منحنی و روش نزدیک‌ترین نقاط صحیح بیان‌شده است.

نتایج حاصل از حذف نویز با استفاده از روش میانگین و فراوانی در همسایگی یک پیکسل

یکی از روش‌هایی که به منظور حذف نقاط صفر از نقشه عمق و جایگزینی آن با عمق مناسب در این تحقیق مورد استفاده قرار گرفته است استفاده از دو روش میانگین و میانه در همسایگی یک پیکسل است. در شکل (6) نتایج حاصل از اجرای این دو روش نشان داده‌شده است. در این تصویر ابعاد پنجره مورد استفاده برابر با 3 است. همان‌طور که مشاهده می‌شود با استفاده از الگوریتم میانگین بیشتر نقاط صفر حذف‌شده‌اند. در حالتی که از الگوریتم فراوانی استفاده‌شده است، تعدادی از نقاط صفر حذف‌شده و بسیاری از آن‌ها همچنان در تصویر مشاهده می‌شود. در هر دو حالت استفاده از الگوریتم‌های میانه و میانگین حذف تمام نقاط صفر با استفاده از پنجره با ابعاد بزرگ‌تر امکان‌پذیر است. ولی در حالت کلی استفاده از این دو روش خطاهایی به دنبال دارد که در ادامه توضیح داده‌شده است.

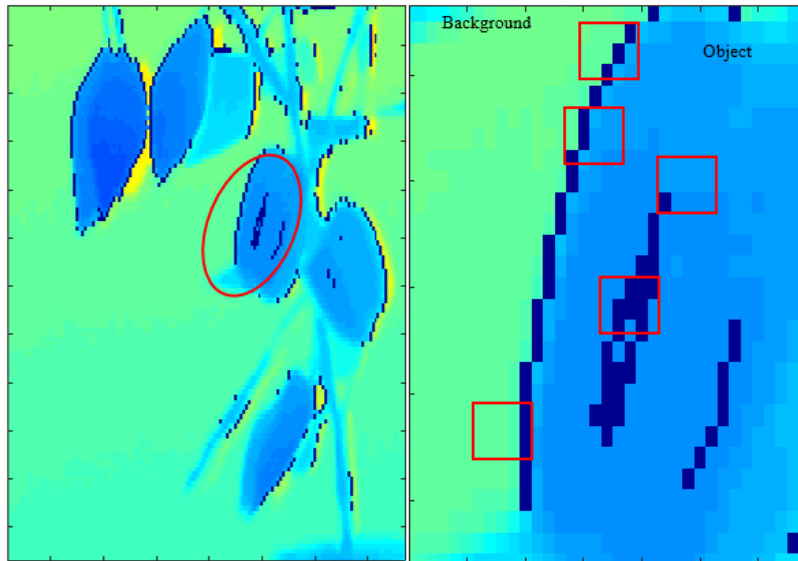


شکل 2. نتایج حاصل از حذف نویز با استفاده از دو روش میانگین و فراوانی

در استفاده از این دو روش حالت‌های مختلفی ممکن است اتفاق بیفتند. همان‌طور که در شکل (7) نمایش داده شده است. هرچقدر تعداد نقاط با عمق صفر در یک همسایگی 5 تایی بیشتر باشد، خطای الگوریتم در محاسبه نقاط صفر افزایش می‌یابد. به‌عنوان مثال در شکل 6a مقدار میانگین در همسایگی برابر با 0 و مقدار میانه برابر با 5 است. در این حالت می‌توانیم بیان کنیم که جایگزینی نقطه با مقادیر میانه و میانگین می‌تواند مناسب باشد. در شکل‌های 7b تا 7c تعداد صفرها افزایش یافته است که هرکدام از این حالات ممکن است در نقشه عمق به وجود آید. بنابراین در مواردی که فراوانی این نقاط افزایش پیدا می‌کند استفاده از الگوریتم‌های میانگین یا میانه روش صحیحی نیست. اگرچه در این تحقیق تعداد لایه‌های صفر کاهش یافته است ولی این کاهش کاملاً وابسته به تکرار الگوریتم است. بنابراین در مواردی که تعداد پیکسل‌های صفر که در کنار هم قرار گرفته‌اند بیشتر از 3 لایه باشند. همه لایه‌های صفر حذف نمی‌شوند. بنابراین احتمال وقوع حالات شکل 7 یا هر شرایط دیگری امکان‌پذیر است. لازم به توضیح است که در مواردی که فراوانی مربوط به اعداد غیر از صفر است نیز ممکن است جایگزینی با فراوانی‌های غیر صفر نیز انتخاب صحیح نباشد. برای بسط بیشتر این مسئله اگر به شکل 8 دقت کنیم مشخص می‌شود که این روش برای همه حالات نمی‌تواند مناسب باشد. چراکه در بعضی موارد ممکن است بیشترین همسایگی نقطه‌ای از پس‌زمینه را شامل شود که به دلیل اختلاف عمق بسیار زیادی که وجود دارد، این روش کارایی خود را در مسائلی که نقاط لبه هدف از اهمیت بالایی برخوردار است، از دست می‌دهد.

a	b	c	d
5 5 3 5 5	0 5 3 5 5	5 0 0 0 0	0 0 0 0 0
5 4 5 5 3	5 4 5 5 3	0 0 5 5 3	0 0 0 0 0
3 5 0 5 5	3 5 0 5 0	0 5 0 5 0	0 0 0 0 0
4 5 4 5 4	4 5 4 5 4	0 5 0 5 0	0 0 0 0 0
5 5 5 5 5	5 5 5 5 5	0 0 5 0 5	0 0 0 0 0
Average=4.4 Mode=5	Average=4 Mode=5	Average=1.29 Mode=0	Average=0 Mode=0

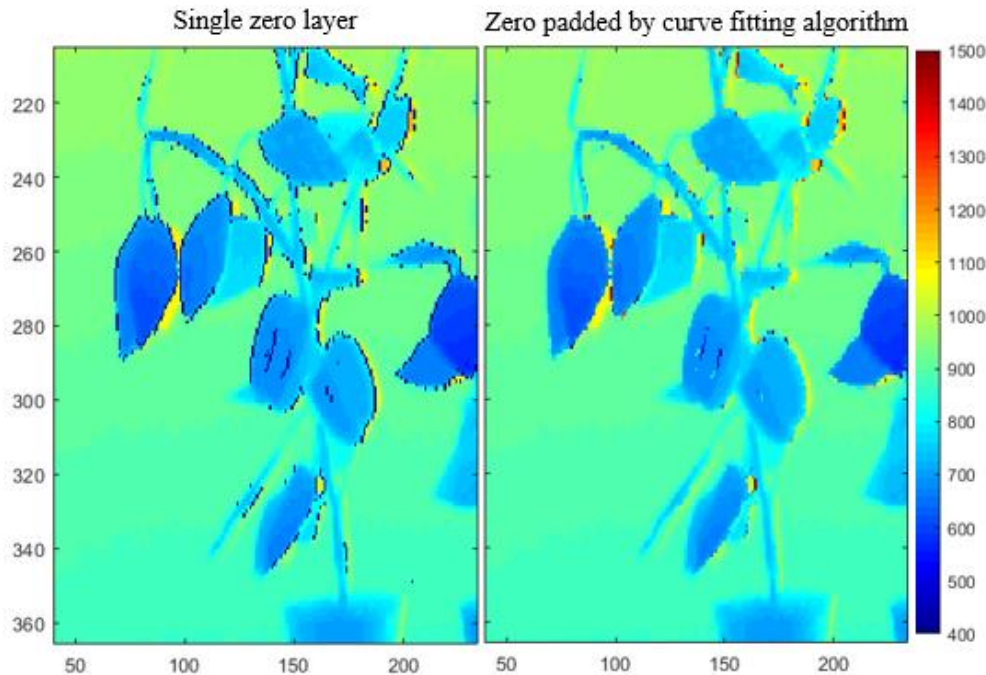
شکل 3. بررسی حالت‌های مختلف در استفاده از دو روش میانگین و میانه



شکل 8. نمایش انتخاب مقادیر در یک همسایگی پیکسل صفر

نتایج حذف نویز با استفاده از روش برازش منحنی (Curve fitting)

در این الگوریتم به منظور تخمین نقاط از روش برازش منحنی استفاده شده است. در این حالت مقادیر 1 تا 4 برای درجه معادله مورد بررسی قرار گرفته است. در هر چهار حالت حذف بیشتر نقاط صفر انجام می‌شود (شکل 9). لازم به ذکر است که در هر چهار حالت حذف نقاط به صورت یکسان از لحاظ شکلی انجام می‌شود. ولی در هر حالت مقادیر جایگزین شده با توجه به همسایگی پس از آن متفاوت است. در جدول (؟؟) نتایج حاصل از این الگوریتم در حالت‌های مختلف مورد بررسی قرار گرفته است.



شکل 9 نمایش حذف نقاط صفر پس از اعمال الگوریتم برازش منحنی با درجه 4. سمت چپ: تصویر با یک لایه صفر در لبه‌های شی. سمت راست: تصویر پس از اعمال الگوریتم برازش منحنی

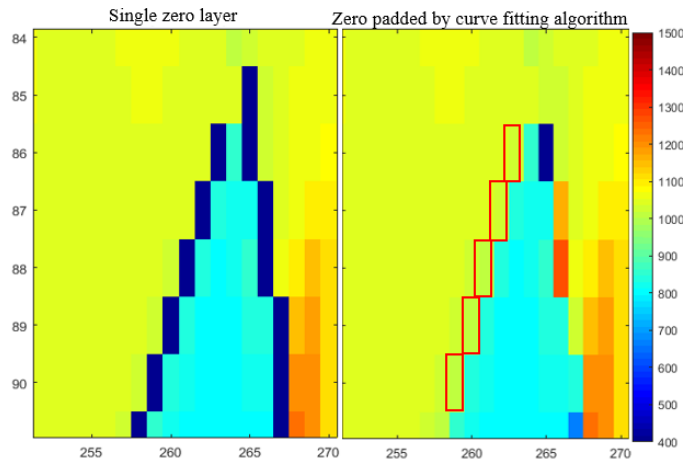


برای این منظور از سه نقطه در نقشه عمق پردازش شده اطلاعات پیکسل جایگزین شده و پیکسل‌های بعد از آن آورده شده است.  $X(i,j)$  نشان‌دهنده پیکسل صفر،  $X'(i,j)$  نشان‌دهنده مقدار جایگزین شده برای  $X(i,j)$ ، است  $X(i,j+1:j+4)$  نشان‌دهنده پیکسل‌های بعدی است.  $d$  مقادیر درجه معادله برازش شده را در همسایگی نقاط نشان می‌دهد.

جدول 1. نتایج عددی حاصل از اعمال الگوریتم برازش منحنی در حالت‌های مختلف

S ample	$X(i,j)$ (mm)	$X'(i,j)$ (mm)	$X(i,j+1)$ (mm)	$X(i,j+2)$ (mm)	$X(i,j+3)$ (mm)	$X(i,j+4)$ (mm)
1	0	1017	839	814	803	802
	0	1008	839	814	803	802
	0	110	839	814	803	802
	0	1015	839	814	803	802
2	0	891	710	686	673	669
	0	891	710	686	673	669
	0	888	710	686	673	669
	0	885	710	686	673	669
3	0	957	703	689	686	686
	0	948	703	689	686	686
	0	948	703	689	686	686
	0	955	703	689	686	686
4	0	1033	853	0	1026	1050
	0	1032	853	0	1026	1050
	0	1030	853	0	1026	1050
	0	1029	853	0	1029	1050

در مواردی که تعداد کمی پیکسل غیر صفر بین دو پیکسل صفر قرار گرفته است (شکل 10 و ردیف 4 از جدول 1). این خطا وجود دارد که مقادیر لبه‌ای صفر با استفاده از مقادیر مربوط به پس‌زمینه تخمین زده شوند. در این حالت قطعاً مقدار تخمین زده شده مقدار صحیحی نیست.

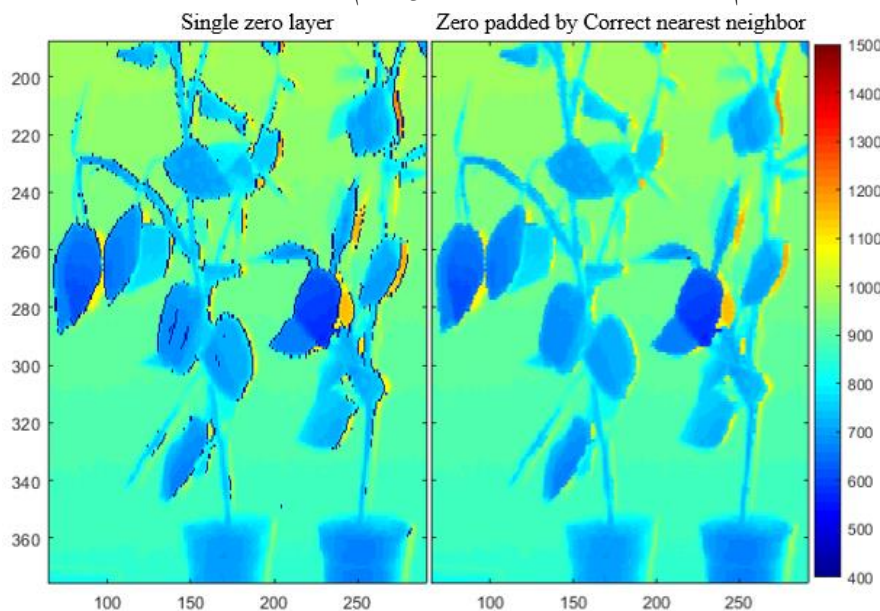


شکل 10. نمایش نتایج حاصل از حذف نقاط صفر با استفاده از برازش منحنی در قسمت‌های باریک هدف.

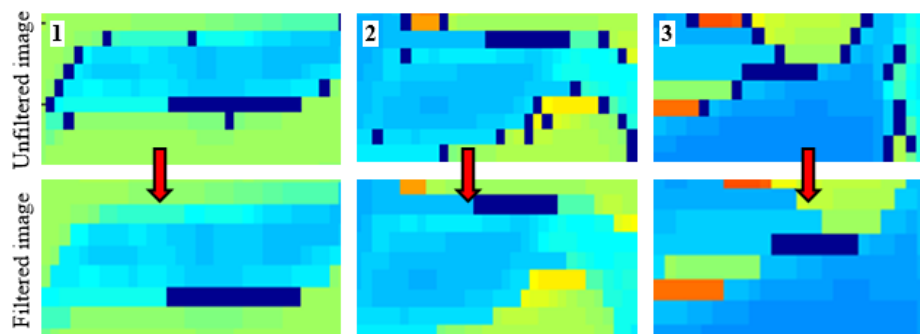
نتایج حاصل از حذف نویز با استفاده از نزدیک‌ترین نقاط صحیح

روش دیگری که در این پژوهش به منظور حذف نقاط صفر معرفی شده است، همان‌طور که در شکل 5 قابل مشاهده است، استفاده از نزدیک‌ترین نقطه صحیح به نقطه صفر است. همان‌طور که بیان شد انتخاب نقاط جایگزین با استفاده از نقاطی که اختلاف زیادی با هدف دارند، در واقع انتخاب نقاط از پس‌زمینه و جایگزینی با نقطه صفر است. انتخاب نقاط پس‌زمینه می‌تواند شامل انتخاب نقاط برگ، ساقه، میوه یا هر عمق دیگری از محیط اطراف به جای یکدیگر باشد. بنابراین برای تخمین نقاط صفر در لبه‌های یک برگ یا میوه بهترین حالت

انتخاب نقاط داخل برگ یا میوه است. در شکل زیر نتایج حاصل از اعمال این الگوریتم نمایش داده شده است. در این الگوریتم نقاط صفر با پیکسل بعد یا قبل از خود در قسمت‌های مختلف گیاه جایگزین می‌شوند. اگر پیکسل صفر در سمت چپ هدف قرار داشته باشد با پیکسل بعدی خود جایگزین می‌شود. چنانچه در سمت راست تصویر قرار داشته باشد با نقاط قبل از خود جایگزین می‌شود. در صورتی که نقاط صفر جزء نویزهای پروازی نباشد قطعاً با مناسب‌ترین نقطه جایگزین می‌شود. در شکل (11) نتایج حاصل از اجرای این الگوریتم بر روی یک نقشه عمق اعمال شده است. همان‌طور که مشاهده می‌شود حذف نویزهای به وجود آمده این روش از کارایی بالاتری نسبت به روش‌های برازش منحنی و میانگین فراوانی در همسایگی نقطه برخوردار است. در این الگوریتم می‌توان بیان کرد که همه نقاط نویز با مقدار صحیح جایگزین می‌شوند مگر در حالات خاصی که تعداد لایه‌های صفر به یک لایه کاهش نیافته باشد. زیرا تنها در این حالات عبارت شرطی الگوریتم برقرار نیست و این به این دلیل است که نقاط بعد از یک نقطه صفر نیز صفر بوده و همواره از عدد قبل از آن کوچک‌تر است. بنابراین با صفر جایگزین می‌شود. در شکل (12) برخی از این حالات نمایش داده شده است. چنانچه بخواهیم این نقاط نیز حذف شوند با افزایش تکرار در الگوریتم حذف لایه‌های صفر، این کار قابل انجام است.



شکل 11. حذف و جایگزینی پیکسل‌های صفر با استفاده از روش نزدیک‌ترین همسایه صحیح (CNN). شکل سمت چپ تصویر نویز دار و شکل سمت راست تصویر فیلتر شده.



شکل 12. خطاهای موجود در الگوریتم CNN

در جدول 2 نتایج عددی حاصل از اعمال الگوریتم نزدیک‌ترین همسایه صحیح برای 10 نقطه صفر نمایش داده شده است. همان‌طور که مشاهده می‌شود همه نقاط صفر به نزدیک‌ترین عدد قبل و یا بعد از خود تغییر یافته‌اند. همان‌طور که در قسمت موارد و روش توضیح داده شد انتخاب نقاط به نحوی است که هیچ نقطه صفر با نقاط مربوط به پس‌زمینه جایگزین نمی‌شوند. به‌عنوان مثال نمونه 1 از جدول 2 نشان می‌دهد که عمق قبل از نقطه صفر (792 mm) مربوط به پس‌زمینه (نقاط با عمق بیشتر) است. نقطه بعد از آن (735mm) مربوط به قسمتی از تصویر است که در لبه‌های آن عدد صفر قرار گرفته است. بنابراین جایگزین کردن عمق 735 mm به جای نقطه صفر صحیح‌تر است.

جدول 2. نتایج عددی حاصل از اعمال الگوریتم نزدیک‌ترین همسایه صحیح

Sample	X(i,j)(m m)	X'(i,j)(m m)	X(i,j-1)(mm)	X(i,j+1)(m m)
1	0	735	792	735
2	0	696	968	696
3	0	796	796	933
4	0	783	977	783
5	0	761	761	1039
6	0	762	762	978
7	0	745	745	962
8	0	753	1059	753
9	0	767	767	1025
10	0	635	909	635

همچنین در جدول 2 اگر نمونه 7 را بررسی کنیم، هنگامی که الگوریتم به نقطه صفر رسیده است با بررسی عمق نقاط قبل و بعد از آن مقدار عمق 745mm را به نقطه صفر اختصاص داده است. در این حالت عمق 962mm که مربوط به نقطه بعد از آن است مجدداً مربوط به پس‌زمینه نسبت به نقطه صفر است. که انتخاب آن نادرست است. در الگوریتم نزدیک‌ترین همسایه صحیح چنانچه در همه پیکسل‌های نقشه عمق در صورت وجود نویز تنها یک لایه نویز وجود داشته باشد این الگوریتم با دقت 100 درصد قادر به حذف و جایگزینی نقاط با مقادیر صحیح است. با توجه به این که این الگوریتم مانند الگوریتم‌های دیگر تابع شرایطی چون وضعیت همه پیکسل‌های همسایگی نیست و صحیح‌ترین مقدار را انتخاب و جایگزین می‌کند نسبت به الگوریتم‌های، میانگین، میانه و برازش منحنی قابل قبول‌تر و از کارایی بیشتری برخوردار است. بنابراین می‌تواند به‌عنوان یک الگوریتم پیش‌پردازش در نقشه‌های عمق و مدل‌های سه‌بعدی مورد استفاده قرار بگیرد. با توجه به این که الگوریتم نزدیک‌ترین همسایه صحیح از دقت و صحت بالاتری نسبت به سایر روش‌های ارائه شده در این پژوهش برخوردار است مقادیر تخمین زده شده برای این الگوریتم در مقایسه با الگوریتم‌های دیگر برای 5 نقطه به صورت تصادفی محاسبه و در جدول 3 نمایش داده شده است. به‌طور کلی هر سه الگوریتم اختلاف زیادی را با الگوریتم CNN نشان می‌دهند، ولی بیشترین اختلاف در الگوریتم mode و بعد از آن mean مشاهده می‌شود. الگوریتم curve fitting نیز در برخی از موارد با اختلاف کم و در برخی از نقاط با اختلاف زیاد این نقاط را تخمین زده است.

اگرچه همه الگوریتم‌ها مورد استفاده در این تحقیق قادر به تخمین مقداری برای نقاط صفر هستند ولی از دقت بالایی برای تخمین در همه حالات برخوردار نمی‌باشند. بنابراین برای ساخت مدل‌های سه‌بعدی استفاده از همه این روش‌ها توصیه نمی‌شود.

جدول 3. اختلاف عمق محاسبه شده در الگوریتم‌های میانگین، میانه و برازش منحنی در مقایسه با الگوریتم نزدیک‌ترین همسایه صحیح

Sample	Difference between algorithms			Algorithms			
	CNN-Mode	CNN-Mean	CNN-Curvefitting	Mode	Mean	Curvefitting	CNN
1	55	55	252	684	758	955	703
2	770	177	146	0	947	624	770

3	761	227	26	0	9 88	787	76 1
4	740	118	11	0	8 58	751	74 0
5	724	11	179	0	7 35	903	72 4

### جمع‌بندی و نتیجه‌گیری کلی

نقاط صفر در نقشه عمق حاصل از حسگر کینکت پردازش‌های بعدی بر روی این تصاویر را پیچیده‌تر و سخت‌تر می‌کنند. برای حذف این نقاط در این تحقیق از چهار الگوریتم فراوانی، میانگین، برازش منحنی، و نزدیک‌ترین همسایه صحیح استفاده شده است. قبل از اعمال هر چهار الگوریتم در صورت وجود لایه‌های صفر متوالی به یک لایه کاهش یافته است تا به نقاط لبه در برگ، میوه و سایر قسمت‌های گیاه نزدیک شویم. همه الگوریتم‌های مورداستفاده قادر به تخمین مقادیر با توجه به همسایگی نقاط صفر یا نقاط قبل یا بعد از آن می‌باشند. الگوریتم‌های مورداستفاده بر اساس میانگین و میانه روش‌های مناسبی برای این منظور نمی‌باشند. چراکه این دو الگوریتم در بیشتر حالات تابع شرایط موجود در همسایگی تعریف شده هستند و شرایط موجود در همسایگی تعیین می‌کند که چه مقداری به جای نقاط صفر تخمین زده شود. الگوریتم برازش منحنی نسبت به دو الگوریتم فراوانی و میانگین از عملکرد بهتری برخوردار است. در این الگوریتم خطاهایی وجود دارد که به دلیل ماهیت تصاویر عمق حاصل از حسگر کینکت به وجود می‌آید. نتایج نشان داد که الگوریتم نزدیک‌ترین همسایه صحیح در مقایسه با سایر روش‌ها از دقت بالاتر و صحت بالاتری برخوردار است.

#### منابع

1. Khoshelham, K. (2011, August). Accuracy analysis of kinect depth data. In ISPRS workshop laser scanning (Vol. 38, No. 5, p. W12).
2. Nguyen, C. V., Izadi, S., & Lovell, D. (2012, October). Modeling kinect sensor noise for improved 3d reconstruction and tracking. In 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on (pp. 524-530). IEEE.
3. Nguyen, V. T., Vu, H., & Tran, T. H. (2015). An efficient combination of RGB and depth for background subtraction. In Some Current Advanced Researches on Information and Computer Science in Vietnam (pp. 49-63). Springer, Cham.
4. Wasenmüller, O., & Stricker, D. (2016, November). Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In Asian Conference on Computer Vision (pp. 34-45). Springer, Cham.